ISSN 0265-2919

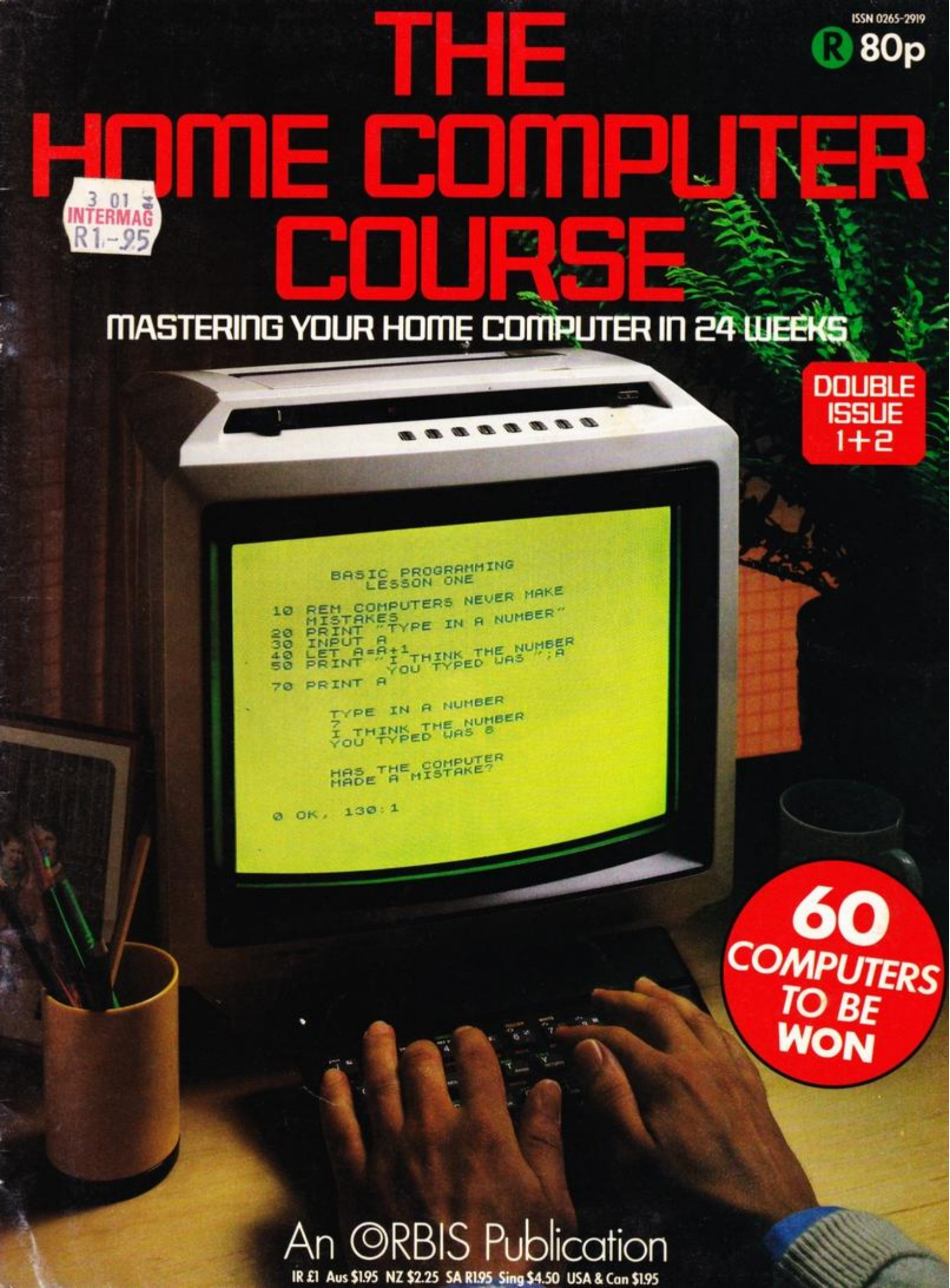
R 80p

THE HOME COMPUTER COURSE

3 01
INTERMAG
R1.-95

MASTERING YOUR HOME COMPUTER IN 24 WEEKS

DOUBLE ISSUE 1+2


BASIC PROGRAMMING
LESSON ONE
10 REM COMPUTERS NEVER MAKE MISTAKES
20 PRINT "TYPE IN A NUMBER"
30 INPUT A
40 LET A=A+1
50 PRINT "I THINK THE NUMBER YOU TYPED WAS ";A
70 PRINT A
TYPE IN A NUMBER
? 7
I THINK THE NUMBER YOU TYPED WAS 8
HAS THE COMPUTER MADE A MISTAKE?
0 OK, 130:1

60 COMPUTERS TO BE WON

An ORBIS Publication

IR £1 Aus $1.95 NZ $2.25 SA R1.95 Sing $4.50 USA & Can $1.95

CONTENTS

What Is A Computer?	1
The Micro Buyer's Survival Kit	5
Setting Up Your Computer	8
Controlling Computers	9
Questions & Answers	13
Games People Play	14
The Small Print	16
Sinclair Spectrum	17
Simply Obeying Orders	20
World Revolution	24
The Electronic Artist	26
Bits And Bytes	28
Oric-1	30
Getting In Touch	32
The Electronic Educator	34
Looping The Loop	37
Behind The Bars	40

Editor
Gareth Jefferson
Managing Editor
Bruce Sawford
Art Editors
Jimmy Egerton
David Whelan
Illustrations Editor
Claudia Zeff
Designer
Hazel Bennington
Art Assistant
Jonathan Sellers
Chief Sub Editor
Catherine Cardwell
Sub Editor
Teresa Bushell
Typesetting
Bunch Typesetters
Office Co-ordinator
Akitaka Kato
Consultant Editor
David Tebbutt
Contributing Consultants
Gary Marshall
Dick Pountain
Contributors
Peter Jackson
Ray Hammond
Tim Heath
Chris Cunningham
Ron Smith
Group Art Director
Perry Neville
Managing Director
Stephen England

Designed and produced by Bunch Partworks Ltd
Published by Orbis Publishing Ltd

Editorial Director
Brian Innes
Project Development
Peter Brookesmith
Executive Editor
Chris Cooper
Production Co-ordinator
Ian Paton
Circulation Director
David Breed
Marketing Director
Michael Joyce

© 1983 by Orbis Publishing Ltd.

Printed in Great Britain by Artisan Press Ltd, Leicester

Welcome to THE HOME COMPUTER COURSE — the first complete guide to the world of home computers, specially designed to help you make the most of your machine.

We'll show you how to speak to your computer, learning how to write programs for it in simple easy stages instead of spending hours being bewildered by the maker's manual or trying to understand the jargon in the micro magazines.

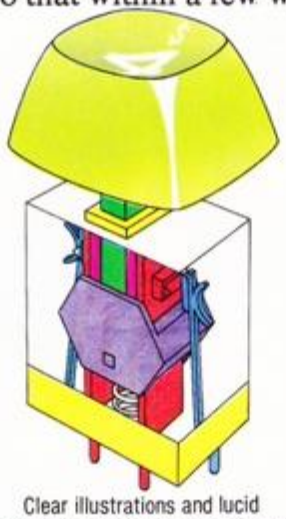
At the same time, you'll be discovering the principles behind writing effective programs — so that within a few weeks, you'll be well on the way to exploiting the enormous versatility of your home computer.

Introduce your child to the fascinating world of computers

We'll also guide you through the maze of software that's available — from the simplest family programs to sophisticated programs for small businesses.

And we'll help make you fluent in the sometimes confusing language that's spoken in the world of computers, explaining the meanings of unfamiliar terms and cutting through the jargon.

And there's more besides. We'll give you all the facts you need to tell one home computer from another. Every week we'll be taking a detailed — and critical — look at a popular model. So, by the end of the course, you'll be able to talk to any dealer with confidence, whether you're buying for the first time or simply looking for ways to upgrade your present system.



Clear illustrations and lucid explanations give you real insights into how computers work

And we'll examine the range of 'add-ons' you can buy, from disk drives and monitors to printers and joysticks.

But THE HOME COMPUTER COURSE isn't all work, by any means. We'll be describing some of the ways computers are affecting the world around us — in our cars and kitchens, from banks to robots and from music to the weather forecast.

Professional computer experts have known for years that computing is fascinating, entertaining — and even profitable.

Welcome to their dynamic world — and to THE HOME COMPUTER COURSE.

A step-by-step guide to computers — from basic principles to complete mastery

HOME COMPUTER COURSE — Price UK 80p IR 1.00 AUS $1 95 NZ $2.25 SA R1.95 SINGAPORE $4.50 USA and CANADA $1.95
How to obtain your copies of HOME COMPUTER COURSE — Copies are obtainable by placing a regular order at your Newsagent, or by taking out a subscription. Subscription rates: For six months (26 issues) £23.80; for one year (52 issues) £47.60. Send your order and remittance to Punch Subscription Services, Watling Street, Bletchley, Milton Keynes, Bucks MK2 2BW, being sure to state the number of the first issue required.
Back Numbers UK and Eire — Back numbers are obtainable from your Newsagent or from HOME COMPUTER COURSE. Back numbers, Orbis Publishing Limited, 20/22 Bedfordbury, London WC2N 4BT at cover price. AUSTRALIA: Back numbers are obtainable from HOME COMPUTER COURSE. Back numbers, Gordon & Gotch (Aus) Ltd, 114 William Street, PO Box 767G Melbourne, Vic 3001. SOUTH AFRICA, NEW ZEALAND, EUROPE & MALTA: Back numbers are available at cover price from your Newsagent. In case of difficulty write to the address in your country given for binders. South African readers should add sales tax.
How to obtain binders for HOME COMPUTER COURSE UK and Eire — Details of how to obtain your binders (and of our special offer) will be in issue 7. EUROPE: Write with remittance of £5.00 per binder (incl. p&p) payable to Orbis Publishing Limited, 20/22 Bedfordbury, London WC2N 4BT. MALTA: Binders are obtainable through your local Newsagent price £3.95. In case of difficulty write to HOME COMPUTER COURSE binders, Miller (Malta) Ltd, M.A. Vassalli Street, Valletta, Malta. AUSTRALIA: For details of how to obtain your binders see inserts in early issues or write to HOME COMPUTER COURSE binders, First Post Pty ltd, 23 Chandos Street, St Leonards, NSW 2065. The binders supplied are those illustrated in the magazine. NEW ZEALAND: Binders are available through your local Newsagent or from HOME COMPUTER COURSE binders, Gordon & Gotch (NZ) Ltd, PO Box 1595, Wellington. SOUTH AFRICA: Binders are available through any branch of Central Newsagency. In case of difficulty write to HOME COMPUTER COURSE binders, Intermag, PO Box 57394, Springfield 2137.
Note — Binders and Back Numbers are obtainable subject to availability of stocks. Whilst every attempt is made to keep the price of the issues and binders constant, the publishers reserve the right to increase the stated prices at any time when circumstances dictate. Binders depicted in this publication are those produced for the UK market only and may not necessarily be identical to binders produced for sale outside the UK.

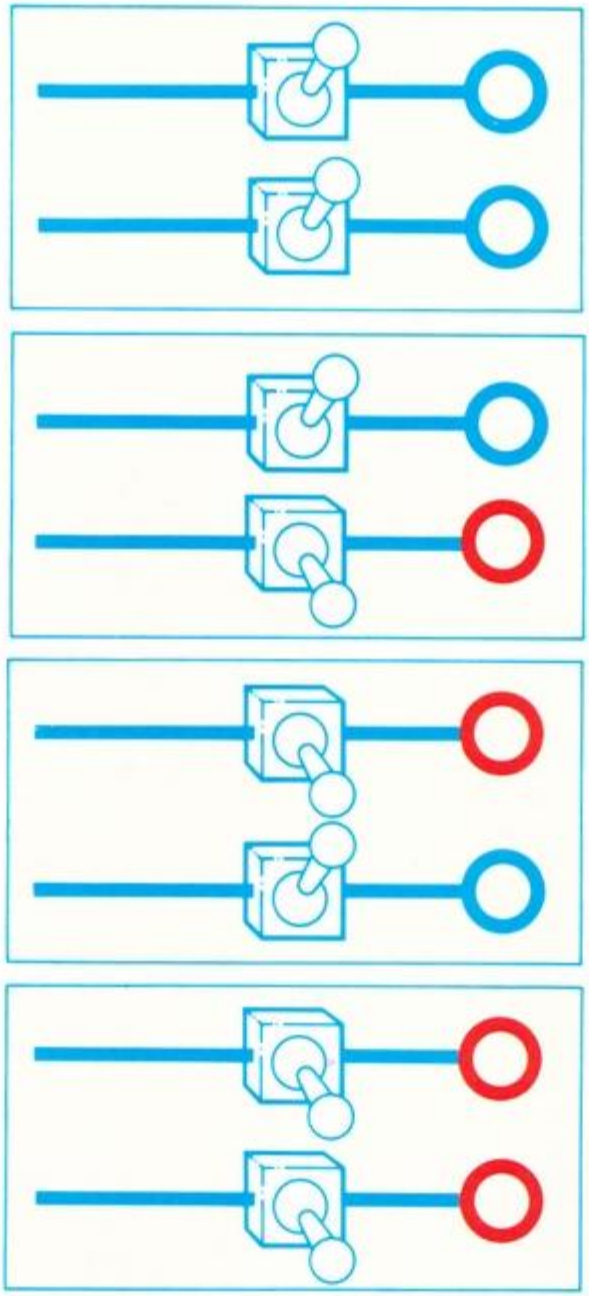
What Is A Computer?

How do computers 'think' and how much do they 'know'? The answers are vital to understanding computers

The question "What is a computer?" is not as easily answered as "What is a television?" or "What is a washing machine?" because the computer, unlike these other appliances, has no single purpose. Digital computers, including those you can buy for £50 in your local high street, are a new breed of machine that can perform an almost infinite variety of jobs, according to the program their owners give them.

The idea of programmability is not altogether unfamiliar to the modern home; after all, many devices such as washing machines and cookers now have a number of different programs built into them so that you can use them in different ways. With a computer, though, the whole function of the machine can be changed by putting in a new program: from a word processor, to an arcade game, to a machine that looks after your accounts, in a matter of minutes.

How does a computer perform so many different tasks? We shall be learning more about that as the course progresses, but first of all let's take a



Switching Into Numbers
Computers use electrical circuits to represent numbers. The circuits consist largely of switches. A switch may be in either of two states; on or off. Two switches together can make four combinations of on and off. Computers use a system like this to represent numbers. Off/off is zero, off/on is one, on/off is two and on/on is three. Using groups of more than two allows larger numbers to be represented. Computers can process large numbers and complicated mathematical operations very quickly using thousands of microscopic switches

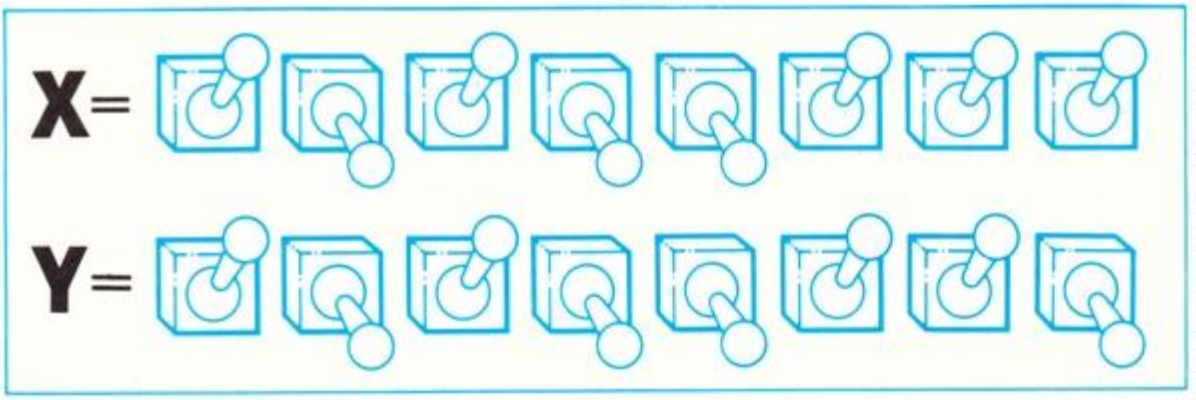
quick look at some of the principles involved.

On one level, a computer is nothing but a box full of tiny electric switches that can be connected together in different ways. This, however, is not the best place to start if you want to understand what computers can *do*; only the men and women who design and build them really need to understand this level, but the rest of us don't. For one thing a modern computer is an exceedingly complex machine; thanks to the astonishing developments in microelectronics (the famous silicon chip) it is possible for even a small home computer to contain some 250,000 of these little switches. All of these switches can be either 'on' or 'off'. Any pools punter will tell you that the number of combinations of 'on' or 'off' is staggeringly large. For another thing the computer you buy will have a program permanently built into it that disguises this mind-boggling complexity, and allows you to 'talk' to the machine using a few shortened but easily recognisable English words.

Many people are surprised when they first use a computer because when they switch it on, they discover that it knows nothing useful at all. Oddly enough, the notion is not yet dead that the computer is an 'electronic brain' that is supposed to know everything. Surely it must know what the capital of Afghanistan is called? Or the height of Mt Kilimanjaro? In fact, far from knowing all these things, the silicon chip that forms the 'brain' of a microcomputer doesn't even know the alphabet or any arithmetic. All it understands are several hundred number combinations, and everything else that it can be taught has eventually to be translated into these numbers. The little switches already mentioned can remember numbers; a pattern of ON and OFF switches represents a number (in the binary number system which only uses '0's and '1's). The fact that the computer can remember, in other words, *store,* information, is vital to the way it works; the electronic memory in a Sinclair Spectrum holds information equivalent to six pages of words in this course (it could store much, much more again on tape, but that is all it can hold on its own).

As well as storing numbers in its memory, a computer can do things *to* these numbers: it can add and subtract them, compare them with each other and move them about inside its memory. Everything that the machine can do is built up from these simple acts. Suppose we want to store text in the computer. Let's invent a code, so that each letter of the alphabet is given a number: then the computer can store words as numbers and shuffle them around. We want to play Frogger? Let's take a picture of a frog and draw it in the squares of a grid so that each little square can be given a number . . . It isn't, of course, necessary to invent these codes yourself because all this work has already been done by the manufacturers and designers and put together in the shape of computer programs.

What is a program? It's a list of instructions to the computer to perform those simple actions (add, compare, etc.) in a particular order, just as a knitting pattern tells the knitter how to perform a sequence of simple stitches in a certain order to produce a garment. But what are these instructions, and how do they get to the computer? Actually they are just more numbers and they are also stored in the computer's memory! This seems to present us with a chicken-and-egg paradox. The computer can do nothing without a program to tell it what to do; every time you press the letter 'A' on the keyboard a program inside the computer must scan the keyboard, find out what key you pressed and then tell the computer the number code for that letter. But at some point, when the computer was first designed, this keyboard-scanning program did not exist. Painstakingly, someone had to put the right numbers directly into the keyboard's memory, using special instruments, just to enable it to understand letters typed on its keyboard and to let it show those letters on a television screen.



A Code For Letters And Numbers

A group of eight switches allows 256 unique combinations of on and off. This is more than enough for an individual code (using nothing more than ones and zeros) for each of the letters, numerals and special signs on a computer's typewriter-like keyboard. The illustration shows how the letters X and Y are represented inside the computer using the ASCII code

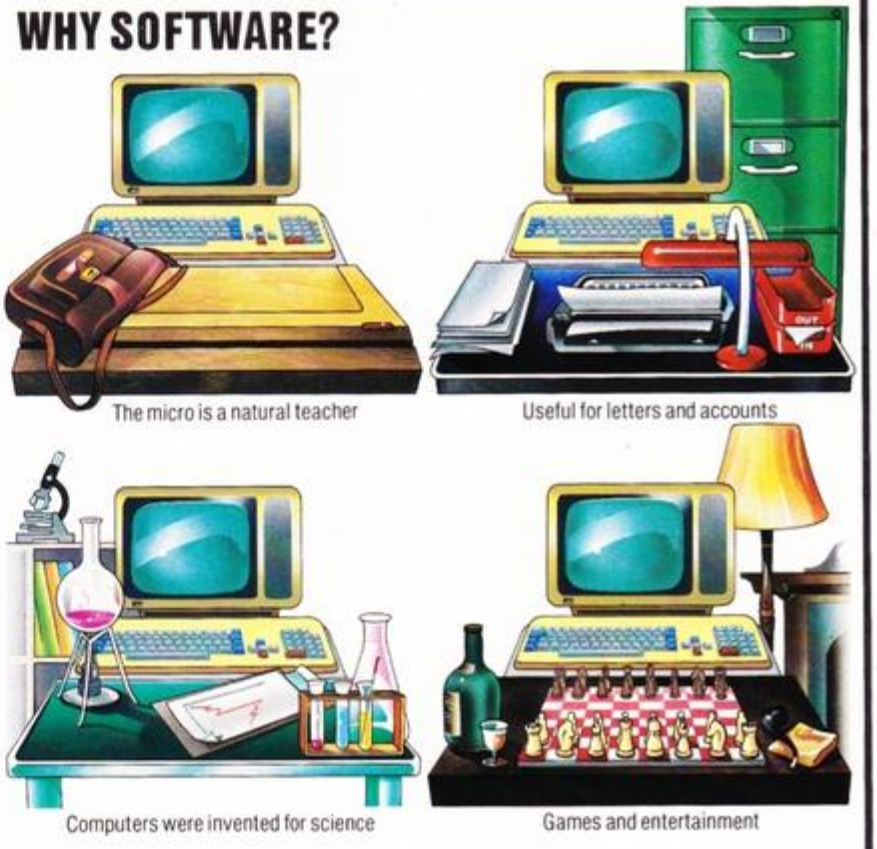
But once those first essential programs are made, everything becomes easier. You can now put new numbers into the computer's memory by typing them in. This process is called machine code programming and we'll be talking more about it in the future. But machine code programming is rather difficult and tedious and so some ingenious programmers have written programs (in machine code) that will translate English words like PRINT, BEEP, LOAD and LIST into machine code instructions that the computer can use. All but the most sophisticated home microcomputers have such a program built into them; as a result you can program them in a simple computer language called BASIC, rather than in streams of numbers. But every time you use BASIC (even if it is only the word LOAD to load up Missile Command) remember that the product of hours of programmers' work is already there inside the computer working for you.

With computer languages like BASIC it is quite easy to write programs to do useful or amusing things, and to be blissfully unaware of all the frantic and complex activity that goes on inside the machine merely to detect that you've typed the letter 'A'. For instance, it is a simple matter to write a program that will store away the names of the capitals in the world and produce to someone's query "What is the capital of Afghanistan?", the answer "Kabul". In other words the electronic brain knows only what you tell it in the first place; it can't discover things for itself.

If this is the case, why are computers so useful? Because they can store vast quantities of information, and they can manipulate it much better than people can. And, of course, putting the information there in the first place needn't always be done by *you*. You might buy a program, written by someone else, with all the world's capitals stored on it; in this case the computer is acting like an electronic reference book. Alternatively, you might buy a program that works upon information that you have typed into it: a 'word processor' that lets you type, correct and redraft documents and letters for instance, or a 'database' program that will let you catalogue a huge library of books and find out answers to questions like "What books do I have by George Bernard Shaw published in London before 1926?" in a few seconds.

The fact that the poor dumb computer understands only numbers is in practice a strength rather than a weakness. If computers actually dealt with the objects that interest us, say words or colours, they would be many times more complex even than they are now, and you would need a different sort of computer to handle each kind of job. How exactly would you store GREEN in a computer's memory anyway? But once the principle is grasped that the computer does not need to 'understand' what it is dealing with, in the way that a person does, then one kind of computer can deal with almost anything. All that is necessary is that a programmer should be able to describe the problem in a way that can ultimately be reduced to

WHY SOFTWARE?



The micro is a natural teacher

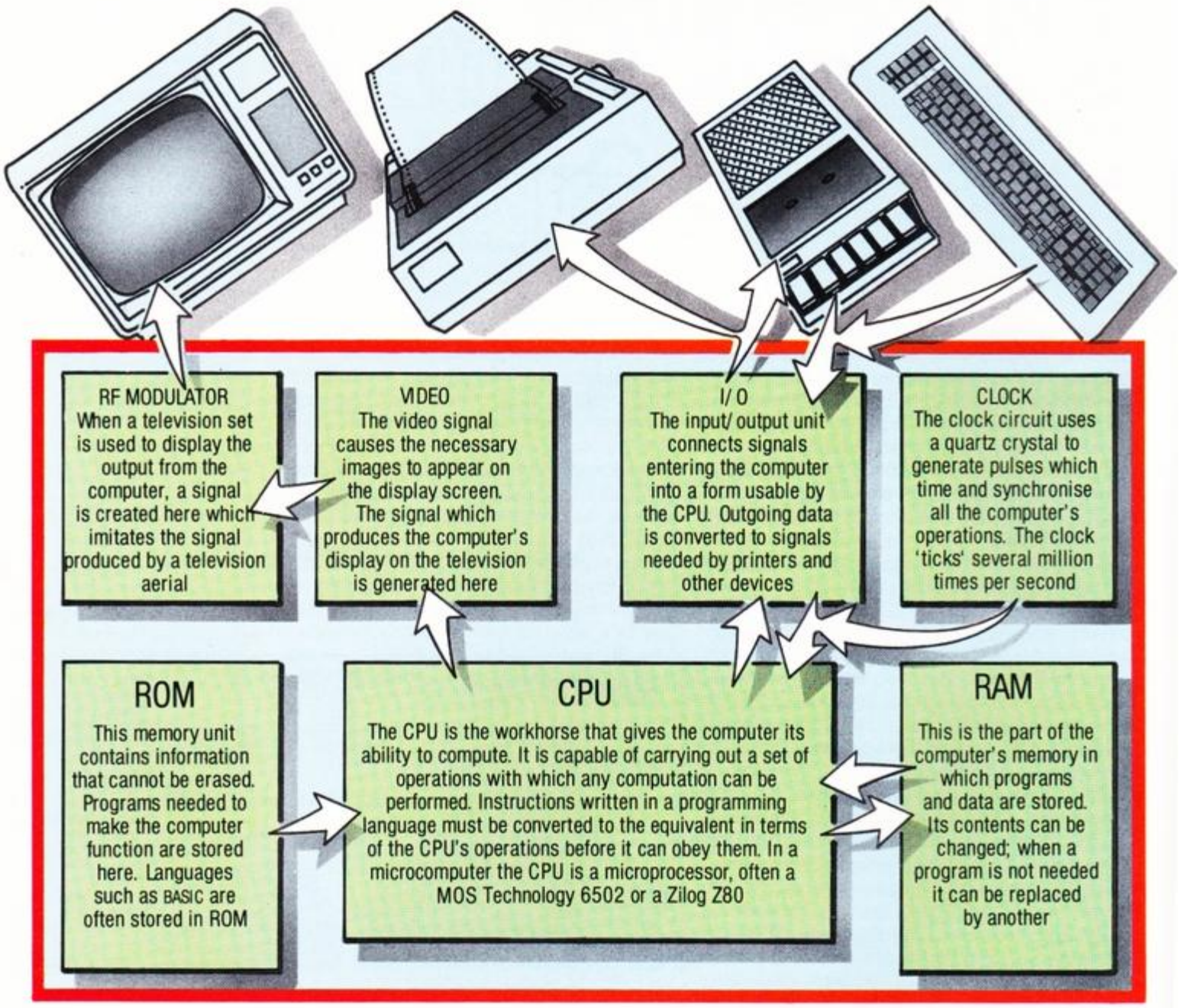
Useful for letters and accounts

Computers were invented for science

Games and entertainment

A computer is a versatile machine and can assume many roles. Software focuses its power. The same machine can be used by the businessman with business software, the technologist using statistics software, or for entertainment by supplying the computer with games software. It is the software that determines what the computer does

TONY LODGE


RF MODULATOR
When a television set is used to display the output from the computer, a signal is created here which imitates the signal produced by a television aerial
VIDEO
The video signal causes the necessary images to appear on the display screen. The signal which produces the computer's display on the television is generated here
I/O
The input/output unit connects signals entering the computer into a form usable by the CPU. Outgoing data is converted to signals needed by printers and other devices
CLOCK
The clock circuit uses a quartz crystal to generate pulses which time and synchronise all the computer's operations. The clock 'ticks' several million times per second
ROM
This memory unit contains information that cannot be erased. Programs needed to make the computer function are stored here. Languages such as BASIC are often stored in ROM
CPU
The CPU is the workhorse that gives the computer its ability to compute. It is capable of carrying out a set of operations with which any computation can be performed. Instructions written in a programming language must be converted to the equivalent in terms of the CPU's operations before it can obey them. In a microcomputer the CPU is a microprocessor, often a MOS Technology 6502 or a Zilog Z80
RAM
This is the part of the computer's memory in which programs and data are stored. Its contents can be changed; when a program is not needed it can be replaced by another

What Goes On Inside

To set up a complete computer system and make it ready for use, it is necessary to connect several units together. The silicon chips that make the home computer possible are packed inside the case, usually under the keyboard. Take the lid off and these are the main components you'll find

numbers. For instance, if we want to make a computer produce music, then we certainly wouldn't expect to have real sounds floating inside it; instead we would describe each note of the scale by a number that is proportional to its pitch or frequency. We can arrange for the computer to send the electrical signals that it uses to represent numbers to a loudspeaker instead of a television screen so that we can hear the results. How do we make a missile shoot across the screen towards the oncoming Space Invaders? Merely move some numbers, which represent a missile shape, from one place to another in the part of the computer's memory that acts as a 'map' of the television screen. Pictures, movement, colour, sound can all be given a suitable number code so that the computer can manipulate them, and a suitable 'transmitter', like a television or a loudspeaker, to turn them back into signals that have meaning for us.

So the final answer to the original question "What is a computer?" must go something like this. It is a machine that stores electronic signals that represent numbers. Some of these numbers are instructions that tell the computer what to do with the other numbers. It will follow these instructions, exactly, without tiring, without making mistakes (though it will faithfully reproduce our own programming mistakes) at the rate of many thousands of operations per second. The end result of these tireless manipulations is yet more numbers. These are 'translated' into the information we want, in a form we can understand. It is the activity of human programmers that makes the computer useful, by exploiting its dexterity with numbers to perform tasks that are meaningful to us; taking in information in various forms and transforming it in ways that would otherwise be too tedious, time consuming or complex.

The Micro Buyer's Survival Kit

Set out to buy a computer and you'll be reeling at the range on offer. Here's how to emerge unscathed

Video recorders, television sets and hi-fi systems all have one thing in common — each performs a specific task. The degree of sophistication between different models may vary. But a stereo system can only reproduce sound, a washing machine washes clothes, and a television set just receives and displays broadcast signals.

A computer is different. A hundred people can buy the same computer and each will find a unique task for the machine to carry out. This is why purchasing a home computer is so different from any other item you have bought before.

When you set out to buy a home computer, several factors require careful consideration. The very first thing to do is to write out a check list of the things you want the computer to do. For instance, you may want to learn the fundamentals of BASIC programming — in which case a Sinclair Spectrum or an Oric-1 may be the machine for you. Alternatively, you may wish to use your home computer to play games, act as a word processor or handle the home accounts — so a BBC or Commodore 64 might be a better choice. Factors such as price and reliability will probably come high on your list.

Your check list should be exhaustive, so that you don't end up with a computer that simply will not do what you want it to.

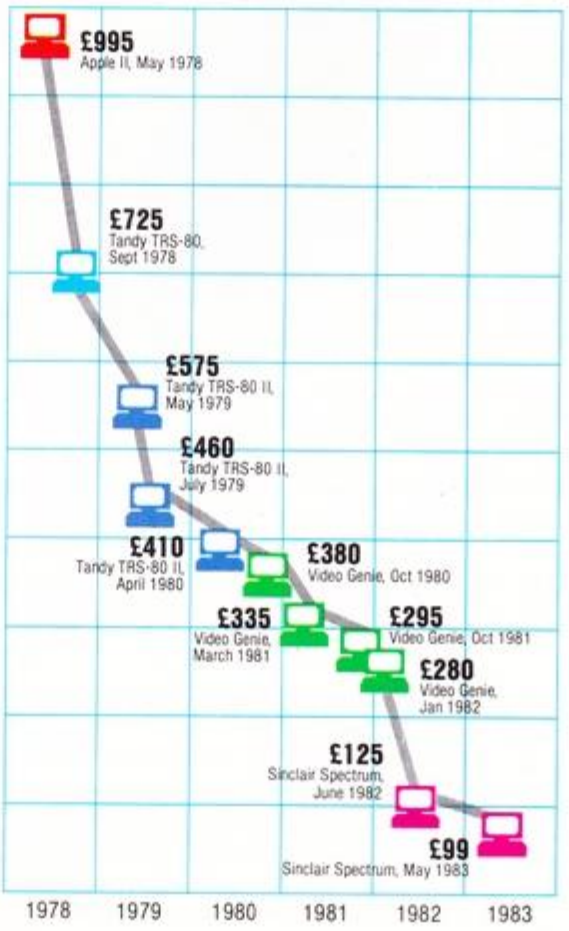
The home computer of your choice may be only the heart of a system. To exploit its full potential you will also require a means of saving programs for future use. A cassette recorder or disk system are typical methods. You will need a television set so that you can see programs and react to games. Often this will involve buying a second set, particularly if you have a family — they won't take kindly to missing their favourite programmes while you are busy with your micro! For anything other than games you may require printed copies of programs or of results produced by the computer, and for this you will need a printer.

If the computer is to be used purely for games, your main consideration will probably be the amount of software that is available. Here micros such as the Sinclair Spectrum, Vic 20 and BBC score highly, as there is a vast and varied number of programs available for them on cassette.

How much memory will you need? This will depend on the complexity of the programs you want to use. The more complex programs will often be larger in size and will therefore require more memory to hold them. Word processors need large amounts of memory to store text. Generally 32K of RAM should satisfy most needs, although 16K will probably be sufficient to run entertaining games software with good graphics. As a rule, go for a machine with as much memory as you can afford.

Some of the more expensive home computers (such as the BBC, Commodore 64, and Atari 800) can be considered suitable for office use. All three have the facilities for adding disk drives, printers, and modems, which link to other computers via the telephone.

So do make sure that you've thought of everything when you've made out your check list. You should be absolutely clear in your own mind what you want from your home computer before parting with your money.



The Fall In Price Of The Micro

These are selling prices from dealers for the most competitively priced computers with at least 16K of RAM. (Prices do not include an add-on RAM pack)

ATARI 400

TYPICAL PRICE: £150

STANDARD MEMORY: 16K

EXPANDABLE TO: Non-expandable

CONNECTS TO: Atari cassette, TV, monitor, Atari disk drive, printer, joystick.

ADVANTAGES: Low cost/Wide range of software.

DISADVANTAGES: Touch-pad keyboard/BASIC must be bought as an extra.

SUMMARY: Touch-pad keyboard and lack of built-in BASIC makes this machine best suited for games.

ATARI 800

TYPICAL PRICE: £300

STANDARD MEMORY: 48K

EXPANDABLE TO: Non-expandable

CONNECTS TO: Atari cassette, TV, monitor, Atari disk drive, printer, joystick.

ADVANTAGES: Typewriter-style keyboard/Wide choice of software/Interchanges software with Atari 400 model.

DISADVANTAGES: The BASIC is long-winded and difficult/Only works with Atari's own cassette or disk drives.

SUMMARY: Has a better keyboard than the cheaper 400 model, and is capable of running business as well as games software.

BBC MICRO

TYPICAL PRICE: A £299, B £399

STANDARD MEMORY: A 16K, B 32K

EXPANDABLE TO: 32K

CONNECTS TO: Cassette, TV, monitor, disk drive, Prestel, Econet to other BBC Micros, Ceefax, joysticks.

ADVANTAGES: Wide choice of software/Clear sound when amplified/Very versatile BASIC.

DISADVANTAGES: Limited colour capacity/Weak sound when not amplified/Manuals difficult to understand.

SUMMARY: This versatile machine can be easily upgraded to a powerful computing tool.

COLOUR GENIE

TYPICAL PRICE: £168

STANDARD MEMORY: 32K

EXPANDABLE TO: Non-expandable

CONNECTS TO: Cassette, TV, printer, monitor, hi-fi, joystick, plug-in cartridges.

ADVANTAGES: Typewriter-style keyboard/Clear sound.

DISADVANTAGES: Comparatively slow operation/Limited choice of software/Poor graphics.

SUMMARY: The cheapest colour computer with a typewriter-style keyboard, but needs an experienced user for its capabilities to be used to the full.

IAN McKINNELL

JUPITER ACE

TYPICAL PRICE: £90

STANDARD MEMORY: 3K+16K

EXPANDABLE TO: 51K

CONNECTS TO: Cassette, TV, monitor.

ADVANTAGES: Includes 16K RAM pack.

DISADVANTAGES: Comes in thin plastic casing/FORTH is not the best language for the beginner/Unreliable keyboard.

SUMMARY: Good for those who wish to experiment with FORTH, and better value now that more software has become available.

LYNX

TYPICAL PRICE: £225

STANDARD MEMORY: 48K

EXPANDABLE TO: 192K

CONNECTS TO Cassette, TV, monitor, printer, disk drive, joystick.

ADVANTAGES: Sharply defined screen display/Built-in machine code monitor/Typewriter-style keyboard.

DISADVANTAGES: Idiosyncratic BASIC/Memory can only be expanded by returning the machine to the manufacturer.

SUMMARY: A well-constructed machine with a keyboard that is pleasing to use.

NEW BRAIN

TYPICAL PRICE: £269

STANDARD MEMORY: 32K

EXPANDABLE TO: 2 Mbyte (2000K!)

CONNECTS TO: Cassette, TV, monitor, disk drive, printer, modem.

ADVANTAGES: BASIC has many features/Capacity for detailed text editing/Semi-portable.

DISADVANTAGES: Non-standard sockets easily damaged/Too complex for the beginner.

SUMMARY: This machine is widely expandable, and is a very competitive lap-held microcomputer. Includes RS232 port.

ORIC

TYPICAL PRICE: £130/169

STANDARD MEMORY: 16K/48K

EXPANDABLE TO: Non-expandable

CONNECTS TO: Cassette, TV, monitor, modem, micro disk drive.

ADVANTAGES: Typewriter-style keyboard/Four-colour printer available/Clear sound.

DISADVANTAGES: Has had reliability problems/Limited choice of software.

SUMMARY: A well-constructed computer that is a good choice for a beginner. The keys are long and narrow, but reliable and pleasing to use.

COMMODORE 64

TYPICAL PRICE: £299

STANDARD MEMORY: 64K

EXPANDABLE TO: Non-expandable

CONNECTS TO: Cassette, TV, monitor, disk drive, printer, joystick.

ADVANTAGES: Very fast screen graphics/Clear sound.

DISADVANTAGES: The BASIC can often be awkward.

SUMMARY: As a home computer the Commodore 64 is good value. It can also be used as a business machine, but this requires expensive add-ons.

COMMODORE VIC 20

TYPICAL PRICE: £130 inc. extras

STANDARD MEMORY: 5K

EXPANDABLE TO: 32K

CONNECTS TO: Cassette, TV, monitor, disk drive, printer, joysticks

ADVANTAGES: Typewriter-style keyboard/ROM cartridge options/ Good quality sound/Wide choice of software.

DISADVANTAGES: Small standard memory size.

SUMMARY: Very popular, but cheaper and equally powerful machines provide more memory. Needs the memory expanded to fulfil its capabilities.

DRAGON 32

TYPICAL PRICE: £175

STANDARD MEMORY: 32K

EXPANDABLE TO: 64K

CONNECTS TO: Cassette, TV, printer, monitor, joystick.

ADVANTAGES: Typewriter-style keyboard/Fast BASIC/Software normally interchangeable with Tandy Color Computer.

DISADVANTAGES: Power socket easily damaged/Unconventional sockets for connecting peripherals.

SUMMARY: Although not outstanding in any way, a fairly priced and very popular machine.

EPSON HX20

TYPICAL PRICE: £486

STANDARD MEMORY: 16K

EXPANDABLE TO: 32K

CONNECTS TO: Cassette, printer, modem, plug-in cartridges.

ADVANTAGES: Typewriter-style keyboard/Built-in display, printer and cassette/Good software and maintenance support.

DISADVANTAGES: Built-in display limited to four lines at a time/Will not connect to a TV or monitor in the UK.

SUMMARY: Among the best-designed of portable computers. It includes an RS232 port and an expansion port.

SINCLAIR ZX81

TYPICAL PRICE: £45

STANDARD MEMORY: 1K + 16K

EXPANDABLE TO: 56K

CONNECTS TO: Cassette, TV, Microprinter.

ADVANTAGES: User definable characters add versatility/Very reasonably priced/Includes 16K RAM pack.

DISADVANTAGES: Touch-pad keyboard/Small size of machine makes it awkward to use.

SUMMARY: This computer lacks sophistication, but can be vastly improved by using a selection of the add-ons available.

SINCLAIR SPECTRUM

TYPICAL PRICE: £99/125

STANDARD MEMORY: 16K/48K

EXPANDABLE TO: 48K

CONNECTS TO: Cassette, TV, Microdrive, modem, Microprinter, joysticks.

ADVANTAGES: Very reasonably priced for a colour computer/Wide variety of games and other software available.

DISADVANTAGES: Very poor keyboard/Barely audible sound/ Temperamental tape load/Gets very hot after prolonged use.

SUMMARY: Very easy to use with an enormous choice of software and add-ons. This is an excellent choice for the beginner.

SORD M5

TYPICAL PRICE: £150

STANDARD MEMORY: 4K

EXPANDABLE TO: 36K

CONNECTS TO: Cassette, TV, printer, monitor, joystick, plug-in cartridges.

ADVANTAGES: Well-constructed/ Power-on indicator.

DISADVANTAGES: Keys too 'rubbery'/Limited BASIC.

SUMMARY: This is an expensive machine to expand. A fuller, and better, BASIC is planned for the future.

TI 99/4A

TYPICAL PRICE: £150

STANDARD MEMORY: 16K

EXPANDABLE TO: 52K

CONNECTS TO: Cassette, TV, monitor, joystick.

ADVANTAGES: Typewriter-style keyboard.

DISADVANTAGES: Extremely slow BASIC/Limited choice of software.

SUMMARY: The lack of software and the slow BASIC make this an unpopular machine. However, at £150 it is reasonably priced.

Setting Up Your System

You need extra hardware for two-way communication with your computer, to store programs — and for some games

Printer

A printer is required when paper copies of programs or printed results from the computer are needed. There are several different types of printer; the price reflects the speed and quality of printing

Disk Drives

Like cassettes, disk drives store programs. Instead of a cassette, a 'floppy disk' is used. Disk drives are much more expensive than cassette recorders but they store more information and work much faster. Generally, disk drives are necessary for business computing

Cassette Recorder

The domestic audio cassette recorder provides a low-cost way of saving programs. The program is stored in the computer's memory while the computer is using it. When the power is switched off the contents of this memory disappear. Before this the program can be recorded on audio cassette tape, and played back into the computer when it is needed again

Television

An ordinary television set allows the computer to display messages. And when you are writing programs, anything you type at the keyboard will also appear on the screen. The monitor shown behind the television is designed to give better-quality pictures with more detail

Joystick

These are similar to the controls found on some arcade games. Their actual use depends on the game being played with them. They might control a spaceship or a character in a maze, for example. Some joysticks have a 'pad' of 10 or more buttons (set out like a calculator); how these are used again depends on the game being played

Track Ball Controller

This is used to play games. By rolling the ball in its holder a game piece can be moved around the screen. It provides much finer, faster, more accurate positioning than joysticks and is more comfortable to use. Buttons are provided for firing 'lasers' and so on

The Computer

The computer is the heart of the computer system, though it needs 'extras' to help it communicate with the user. It has a keyboard similar to that on a typewriter, but with some extra keys. Several sockets are provided (usually on the back of the computer) to connect it to other machines such as the cassette recorder or disk drive and the television set

Controlling Computers

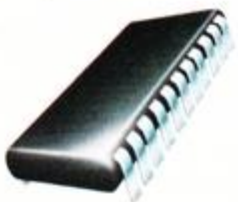
Your computer 'hardware' won't run without the aid of appropriate 'software'. We explain this crucial term — and how to assess the software you find in the shops

Software is the invisible half of a computer system. Without software, the computer is no more than an inert mass of electronic machinery. Without software, the computer can do literally nothing.

Peek inside the silicon chips of a computer and you will find they consist of thousands, perhaps millions, of microscopic electronic switches. Just as a light switch cannot turn a light on or off by itself, the switches in a computer need to be made to turn on or off. They don't all turn on or off together, however. Each individual switch needs to be specifically turned on (or off) and in exactly the right sequence in relation to all the other thousands of switches. Software is how that is done.

Software is the name given to the instructions which make the computer work. These instructions are in the form of numbers which, when presented to the CPU (the heart of the computer; see page 4), set and reset the internal switches to cause specific things to happen. These numbers are only 'understood' by the computer when they are in so-called binary form (converted into ones and zeros as explained on page 28).

These ones and zeros which the computer understands (in the sense that they make it perform specified tasks) are the end product of a long chain of events that started as ideas in the mind of the program writer. A computer program ('program' is the word for any particular single piece of software) can exist in many different forms. The only definite thing we can say about any program is that it must end up in the form the computer understands. Let's take a specific example. Suppose a traffic engineer wants to control a set of traffic lights using a computer. To do this the controlling computer will need a program to make it instigate the correct sequence of events (it's no good having all the lights on green at the same time!). But before this software can be written, the engineer has to think carefully about what exactly it is he wants the computer to do. Usually, these

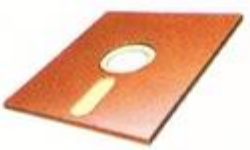


ROM

ROM (Read Only Memory) is one of the main kinds of computer memory devices. A product of the silicon chip revolution, it allows computer programs to be stored permanently. Most home computers are supplied with a ROM chip containing the BASIC programming language. Other ROMs can be bought for some computers to upgrade their performance by adding another language. Word processor ROMs, which turn the computer into an 'intelligent typewriter', are also available

CASSETTE

Software is often supplied on cassette tape identical to that used in recording sound. Games programs usually come in this form. A program is fed from the tape into the computer by connecting the machine to an ordinary cassette recorder and 'playing' the program cassette. The tape is stopped when the program has been loaded and the computer normally does not need to 'look' at it again



FLOPPY DISK

Software (programs) can be stored by recording it on a disk of magnetic film. The recording is made in 'tracks' on the surface, like the bands on an ordinary LP, by a magnetic 'read/write' head, which also 'reads' (plays back) the program when required. Disks offer a large capacity and a high speed of operation, which have to be paid for: they need sophisticated 'disk drives' (see page 8), which are expensive

CARTRIDGE

A cartridge is essentially a ROM packaged in a convenient housing. Some home computers have readily accessible sockets into which these cartridges can be plugged. The software that comes on cartridges tends to be either a programming language (such as BASIC) or sophisticated arcade-style games

To make your computer work, it needs to be 'fed' with software (a set of electronic instructions). The devices pictured here are the 'media' on which those instructions can be stored. They represent the four commonest ways in which software is supplied. Each has its own special advantages. Software is tailored for each make of computer — a program written for one make will not necessarily work on another

would be written down using ordinary English sentences (e.g.: At this point I want signal number one to turn on the orange light whilst keeping on the red light. Then I want both the red and orange lights to go off and the green light to come on). Clearly, these sentences are not in a form any computer could understand so they need to be converted into a program. He uses a programming language such as BASIC. A language of this kind allows logically arranged thoughts (in English) to be rewritten in a way that the BASIC interpreter can understand. The BASIC interpreter is itself a program which converts the original program (written in BASIC) into the form understood by the computer's central processing unit (CPU). Software in this form is called 'machine language' or 'machine code'.

The software you actually buy to use with your computer will always be in machine language and it is stored in a form readily accessible to the computer. Sometimes the software is stored in ROM memory inside the computer. More commonly it is supplied on cassette or on floppy disk. These objects are not the software itself, simply the 'media' in which the software is supplied. To be used by the computer, the software has to be transferred from the cassette (or floppy disk or ROM) into the computer. Once these instructions have been loaded (as the process of transferring the software is called) the program can start to operate.

Buying Software

With a couple of thousand pounds in the bank, you might possibly say to yourself "I think I'll buy a car". It's most unlikely anyone would say "I think I'll buy a machine" because the obvious question would be "What kind of machine? What's it supposed to do?"

It's the same with software. A computer by itself is inert, but the software you buy to use with the computer is capable of turning it into a home arcade game, an automated typewriter or an in-house accountant. So the first thing to decide is what you want your computer to do for you.

Start with the problem and then find the software that provides the solutions. In the search for the right piece of software, there will naturally be a refining process as you analyse your actual needs. If the starting point is how to entertain the kids on a Sunday afternoon, the next stage is to find out what kind of programs are likely to provide that entertainment. Computer games range from arcade-style slaughter of aliens to complex and challenging fantasy simulations (see page 32). If an arcade-style game is what you want the computer to provide, the next question is whether or not it's available for your machine.

Since the differences between computers are more than skin deep (each computer has its own electronics inside and requires individually written software) there is virtually no compatibility between models. A program that works on the

FIRMWARE

The origin of the term 'hardware' is obvious: it's the physical and electronic part of the computer — the power supply connections, keyboard, silicon chips and so on. In contrast, software takes its name from its intangible nature, since it consists simply of a set of instructions. Computer experts also talk about 'firmware'. In the early days of computing, 20 to 30 years ago, software was coded and stored on punched paper tape of the kind familiar to telex operators. Then cassettes and magnetic disks took the place of paper tape. In the 1970s a new technique for storing software in ROMs (purpose-designed) chips — see page 4) was invented. ROM chips have the software instructions built into them during the manufacturing stage. It is this combination of 'intangible' software and 'concrete' hardware that is called firmware

Handle Words

Word Processing

With word processing software, your computer takes you one stage beyond the typewriter. Even good typists make mistakes, but with a word processor you can have perfectly printed letters every time and increased productivity too.

The computer keyboard takes the place of the keys on the typewriter; the television screen substitutes for the paper in the typewriter. The words you type appear instantly on the screen, just as they do on the paper in a typewriter. But there the similarities end, and the power of the computer takes over.

Mistakes can be corrected instantly — on the screen. Words can be retyped or made to disappear. Even whole paragraphs can be deleted. Word processors do more than just delete words, though. If your thoughts could be expressed better by rearranging sentences, you can do exactly that, right there on the screen. The words or sentences you want to move around the 'page' are temporarily deleted (the word processor program takes them off the screen and stores them inside the computer's memory). They can then be inserted exactly where you want them.

When the document has been written exactly the way you want, it can be printed using the computer printer, or it can be stored on cassette or floppy disk for later use

Balance The Books

Accounts Package

Since computers can handle mathematical operations, it is hardly surprising that many programs are available to help the businessman. The range of accounting software is impressive, from automated bookkeeping to full accounting. Programs like these usually have to handle large amounts of information and need to store large numbers of records. Consequently they usually require at least one floppy disk drive to cope with the large storage requirements.

Accounting programs generally work through a system of questions (displayed on the computer screen) and answers (supplied by the computer operator). The information typed in by the operator is manipulated by the computer program, all the necessary calculations are done and the results are stored on the floppy disk or printed on the printer as appropriate.

Such programs include the automatic issuing of invoices, reordering of stock, keeping ledgers and keeping track of work in progress. Prices of software range from about £50 to well over £1000 per program. Such expensive software may be a good investment for a business as it saves on labour costs and gives quicker results

Filing

Databases

Computers can search through files of information far quicker than people can; the more massive the amount of information you need to search through, the more a computer can help. At its simplest (and cheapest) a database may be little more than a computerised address book that can look up names, addresses and telephone numbers. More sophisticated and expensive database programs can perform far more complex operations.

To give an idea of the power of a database, consider a botanist who is compiling information for a book on exotic and poisonous mushrooms. He will have built up extensive files on various species and their habitats. He may also have notes on a wide variety of reference books, and an endless list of individual specialists.

Before the days of an affordable computer, this information would have been written out on cards and filed in a card index system. With a database program and a computer, the information can all be stored in the computer's memory. Using the power of the database, the botanist can get instant answers to his problems. If he needs to have a list of all the fungi ever recorded in Sussex, the database can give it to him. If he needs a list in alphabetical order of all the books containing the word 'poison' or 'poisonous' and 'mushroom', 'mushrooms' or 'fungi', the database can give him that too.

Databases need to handle massive amounts of information and are usually available only on floppy disks. They tend to be expensive, with prices ranging from £50 to over £500 per program

Handle Numbers

Spreadsheets

The spreadsheet is the computer's answer to all those 'what if' questions that used to be tackled with a calculator and reams of paper. Any business with a product to sell has many variables. Changing any one of them will generally affect most of the others.

Consider the questions a cinema proprietor might ask. "If all the seats were sold, how cheap could we make the seat price?" or "Would we get more revenue by reducing the price of ice cream with the same number of usherettes, or should we increase the price and employ two more people?". Each decision is likely to affect the entire business — lower prices may mean increased sales but lower profits. A spreadsheet is a special program that can give instant results to questions like these

All the essential numbers to be manipulated are arranged in a grid of rows and columns and the relationship between each row and column is specified (for example, the numbers in each row of column C is the result of subtracting the number in column A from the number in column B). Once all the real and hypothetical data is assembled, any single figure can be altered and the 'impact' on all the other numbers can be seen instantly.

The people who use spreadsheets are usually businessmen working out costings or engineers and scientists with very variable numerical data to manipulate. Spreadsheets range in price from £30 to over £500 and usually require both disk drives and a printer

Entertain

Play Games

Computers are not only good for processing numbers and words. They can also provide many hours of entertainment if used with one of the many games programs available. These cover a wide range from chess and backgammon to arcade style games and simulations (such as 'lunar lander' and flight simulators). There are also extraordinarily complex adventure games that can take days or weeks to play (see page 14). Many computer games are not only fun, but have considerable educational value too.

Computer games are highly interactive. In other words, they require constant attention and input from the player. This input is usually via the keyboard; a key might be used to fire a 'laser' or a 'missile' or to control the movement of something on the screen. The number of keys used will vary, depending on the game being played, and how much control the program requires.

A popular alternative to keyboard input is the joystick. These are plugged into the computer and operate somewhat like aircraft joysticks. They give greater control, and make playing computer games even more fun

Ready-To-Wear Software

Most off-the-shelf software is sold as a disk or cassette with a manual describing how to use and get the most from the program. Apple Writer is a typical word processor program, costing £120. It comprises a single floppy disk and a comprehensive manual explaining how to use the program. The manual includes a quick tutorial so that beginners can start to use it right away. Standards of documentation vary enormously. Some software comes with manuals so incomplete and badly written that the software may be difficult or impossible to use. It's an important point to watch out for when shopping for software

Atari 800 will not work on the Spectrum (unless a special Spectrum version has been produced), so you have to buy software that has been produced specifically for your machine.

Even now you're not ready to make a purchase. The next considerations are the physical limitations of your machine. Check how much memory your computer has. If it has 16K of ROM see if the game you want needs extra memory to be added. As a rule, the more interesting and sophisticated games require longer programs, so you will need more memory in the computer. And don't forget that software comes in a number of different physical forms (see previous page). If a program is supplied only on floppy disk, but all you possess is a cassette player, you will not be able to use it without first buying a costly disk drive. Some software (particularly games) requires other extras such as joysticks. You are not likely to need a printer for your games, but business software frequently requires one in order to print results.

Finally, there's how much you can afford. Games on cassette start at under £5, but prices can rise rapidly after that. Some business packages on disk cost hundreds of pounds.

Types of Software

In some ways, games are a class apart. The function of a game, after all, is to entertain. Most other software is designed to make a particular job easier and quicker. The ways in which writers of software have managed to increase profitability and efficiency are legion. Consider the poor copy-typist whose ability doesn't satisfy the boss. With a microcomputer and a piece of software called a word processor, the computer takes the place of the typewriter and the corrections are all made on screen. Once all the words are seen to be correct, the whole page can be printed out on paper, automatically and at the touch of a switch. The savings in time and frustration are enormous.

Another tiresome task that lends itself to computerisation is financial administration. Many of the activities that used to keep armies of clerks busy — working out salaries and balancing the company books — can now be performed by specially written software. The programs themselves are quite specialised, so it's unlikely that a single piece of software will answer all of a business's needs. Categories include 'payroll' programs to calculate wages and print pay slips, 'stock control' programs to keep tabs on what goods have been sold or used (sometimes the program can order new stocks automatically) and there are even programs to help work out the most economical sizes and qualities of paper on which to print books and magazines.

Another task computers can do spectacularly well is to file and sort information. This type of program is called a 'database'. Databases can replace whole filing cabinets and can do all the arranging and cross-referencing for you.

The final broad category of software is that known as the 'spreadsheet'. A spreadsheet program allows complicated budgets or financial forecasts to be laid out and endlessly tinkered with, replacing reams of paper and the familiar calculator.

All the types of software we have been talking about are sold 'off the peg'. They are ready-made in the sense that the original writer had a specific set of solutions in mind for the problems as he or she envisaged them. There may come a time, however, when no piece of commercially available software will make your computer do exactly what's needed — and then what do you do? One solution, albeit an expensive one, is to hire a computer programmer to write a program exactly tailored to your needs. The other way is to learn how to write programs yourself. Armed with a language such as BASIC it is possible to generate programs that make your computer do all kinds of amazing things. And the only expense is the time it takes you to write the program.

Questions And Answers

Questions about computing that often spring to mind but are rarely answered in the manuals and magazines

What can a home computer be used for, apart from playing games?
Home computers are used to run small businesses, handle accounts, do word processing, and can even keep records of golf club members and their handicaps, or help in designing interiors. These are fairly typical of the uses to which home computers are put. More generally, the uses can be classified as handling numbers, handling words, storing information and displaying information in a way that suits the user.

Computers seem to be getting smaller and cheaper. When and where is this process likely to stop?
Computers are getting smaller as technology continues to advance. The electronic components inside are getting smaller, but the keyboard cannot drastically change its size. It must remain large enough to be used by our fingers. For this reason, computers are unlikely to become much smaller until the traditional keyboard is replaced by other ways of communicating with the computer. With several computers costing less than £100, it is unlikely that prices can drop much below this level.

Is BASIC a difficult language to learn?
BASIC itself is not at all difficult to learn. By comparison with English, which after all is a language that we have all learnt, BASIC is a small language with rigid rules. This makes it much easier to learn than a foreign language. Although the elements of BASIC are easy to learn, it is not so easy to write very long and complicated programs.

When should I start learning BASIC?
The best time to start is when you need to. It may be that the computer can be made to do exactly what you want by running a program that you can buy. In that case, you don't need to learn BASIC at all. Unfortunately, programs you can buy do not always do exactly what you want. By learning a little BASIC you can sometimes adapt them to your needs. In the end, though, to make the computer do all the things you want, so that it really is a personal computer, you need to start learning BASIC as soon as possible.

A monitor seems to be more expensive than a television. What extra do I get for the money?
You get a much clearer and crisper display on its screen. If you expect to spend any length of time using your computer, a lot of it will be spent looking at the screen, and you will be less likely to get a headache if it is clear and easy to read. Besides this, any graphic images you create with the computer will look better on a monitor.

How much electricity does a home computer use?
Less than a 60 watt light bulb.

The advertisements seem to tell me that I can use my computer to help with my children's education. Is this really true?
Yes. They can also help educate adults. The Department of Education and Science has done its best to ensure that every school in the country has a computer, so they are bound to play an increasingly bigger role in education. The key to using them as a learning aid lies in having suitable software. There are many drill programs to teach multiplication tables and spelling, for example. This is not a particularly stimulating way to use the computer, however. A more imaginative educational use is through the LOGO language. LOGO allows children to learn by exploring a so-called microworld and by carrying out experiments to see what happens. Here, the child learns by actually programming the computer. For more on computing for children, see page 34.

Some computers, like the Spectrum, have rubber 'calculator-style' keyboards; others, like the Dragon, have 'typewriter-style' keyboards. What difference does this make?
Very little, unless you are a trained typist, in which case you can type in your BASIC programs at great speed on a 'typewriter-style' keyboard, but not so rapidly on the other kind.

Games People Play

Dungeons and dragons, stock markets and space flights, excitement and education — all are in computer games

Most people play games on their computers, and the world of computer games is a fascinating kaleidoscope of excitement, puzzling problems, and new challenges.

The video games machines in the arcades and in the home are being overtaken by cheap and powerful home computers that give more variety and thrills without taking all your savings while you learn how to play. These home computer games have introduced more people to the fun of computing than any business accounting software ever did or will. There's no need to feel you're misusing the machine. Games are there to be enjoyed.

Adventure

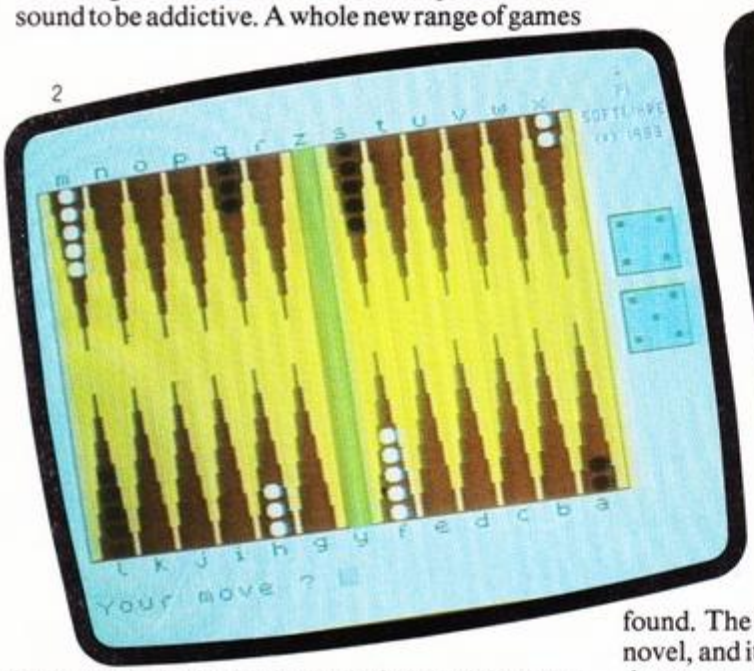
Not all games need to have colour pictures and sound to be addictive. A whole new range of games has appeared with the spread of cheap computer power; games that stimulate the imagination with words in the way that books have always done.

These are called 'adventure' games, after the first program that was written for programmers to play on their giant mainframe computers in their spare time. The idea is that the program creates a world that the player explores by guiding an alter ego; but the guiding is done with words typed on the keyboard rather than with joysticks.

The little character inside the computer's world is moved by typing directions like 'north' and 'up', and the computer gives word pictures of the surroundings and any objects lying around. The player's character can 'take', 'drop', 'turn' or 'break' the objects, or can try any action that might seem helpful. The player can try anything. The only limit is the imagination.

The world inside the computer program can be a maze of caves and dungeons packed with treasures and monsters, as in the popular 'Dungeons and Dragons' role-playing games. Or it can be a deserted alien spacecraft, or even a country house where a murder needs to be solved.

Whatever the scenario, the player has to explore, find useful objects and treasure, and solve intellectual puzzles. The computer world needs to be mapped, and a full score comes only when all problems have been overcome and all treasures found. The best adventure games are like a good novel, and it can take much longer to complete one than to read the book.



Board And Table

It was natural for the old favourite board and table games to be transferred to home computers as soon as the technology could handle it. There is no need to find an opponent when the computer can handle that job, and if you make a wrong move you can correct it without the computer accusing you of cheating. The machine can also improve your game by pointing out and correcting any mistakes you make during play.

Computer chess has reached a very good stan-

dard, and home computer graphics can now produce complete and detailed board layouts with smooth piece movement. Draughts has been completely analysed, and a computer program could easily be the world champion. Backgammon, Bridge, Othello, Pontoon and Go have all been made available on a variety of home machines. Computers make strong opponents on board, table or screen.

Learning Through Play

Several software companies specialise in educational games, with the computer setting tests and puzzles and providing the reward of a game if the answer is right. Or a game may just look like a game, but also have educational information embedded in it about adding up, spelling, or even the law of supply and demand.

A favourite game is one in which children guide a pen-holding robot, known as a 'turtle' because of its shape, over a sheet of paper. In this way they have fun drawing pictures and are learning geometry at the same time.

Arcade-style

Arcade-style games are games for fast action and movement and have drawn billions of coins through the slots of video machines. On home computers you can play all the arcade favourites, with swooping invaders, hopping frogs, digging miners and giant gorillas.

But the software companies have games ideas of their own and have come up with games that match the arcades for excitement and spectacular graphics. With a home computer there is a wider choice of fast and thrilling arcade-style games, all ready to be played when you want, and with no hungry cash slot gaping. These games test home computers — and programmers — to their limits.

Grand Strategy

Games that require thought and planning, the deviousness and single-mindedness of a great general, have also moved onto the micro.

The player can be any general in any war, deploying armies and trying to out-think and outplan the computer opponent. The computer is at its best acting as the umpire and the board controller, as it cuts out the densely written rules and easily lost cardboard pieces that have kept board wargames a minority cult.

Alternatively, the player can be the king of a small country, working out how to husband the crops and treasure while keeping the workers happy and fed and keeping robbers out of the fields.

5

1 Wizardry: Step into a mysterious world. Your companions can help. But who to choose? A warrior, a maiden or a scientist?

2 Backgammon: Your opponent in this game is backed by a formidable power — the relentless logic of the computer

3 ABC Dragon: An educational game for children

4 Zaxxon: One of the original arcade games. The screen becomes the pilot's windscreen as he weaves his way through missiles and dogfights to his goal!

5 Eastern Front: Perhaps you can succeed in Russia where Hitler failed

6 Aircraft flight simulation: Fly the plane or crash!

Equally, the player can be in charge of a country's energy supplies, weighing the costs of coal, oil and nuclear power against their dangers and long-term effects. The computer can help you take the long view, and perfect your world takeover schemes.

High Flying

Games programs can put you in the cockpit of a light aircraft, reading the instruments and handling the controls to make perfect take-offs and landings at a variety of simulations of real airports; they can make you pilot of a space shuttle mission, complete with views of the Earth through the portholes; or they can make you a Rockefeller-style tycoon on the stock markets of the world, making and breaking the giant corporations.

Accuracy is all in simulation games. Follow the rules of the real world, and the game will show you what would really happen. But make a mistake, and you don't find yourself trapped in tangled wreckage or having to take a plunge from a Wall Street balcony. Computers are more forgiving than the real world!

The Small Print

You can't judge a book by its cover — but you can tell a lot about a micro by its technical specification

Keyboard

The keyboard is specially designed to be easy and pleasant to use. Its keys have the standard typewriter-style layout. The character on any key can be displayed repeatedly by keeping the key depressed. Capitals and ordinary letters can be displayed and a separate group of keys (numeric keypad) is provided for entering numbers

Display

The ASCII (American Standard Code for Information Interchange) character set is a standard set of letters, numbers and symbols used by many computers. On some computers the screen displays these characters in 80 columns and 25 rows. The picture can be shown on a television or a special monitor

Memory

The numbers give memory capacity in kilobytes, or thousands of bytes. ROM (Read Only Memory) contains the facilities needed for the fundamental operation of the computer, usually including a language such as BASIC. RAM (Random Access Memory) is for storing the user's programs and data

Interfaces

There are special sockets through which a printer, communications equipment, a cassette recorder and cartridges can connect to the computer. A cartridge is a special ROM which can contain a program, a language or even a new O/S (operating system)

CPU

The CPU is the Central Processing Unit — the silicon chip that is the heart of the computer. This one, a Zilog Z80 microprocessor, is one of the most common. The clock that times all its operations can measure as accurately as 2.2 million times a second

Features of "TYPICAL" Computer

Memory	16 Kbytes ROM, 32 Kbytes RAM, capable of addressing 48 Kbytes RAM
Display	Can display ASCII character set 25 rows each with 80 character positions, outputs to domestic TV & monitor
CPU	Z80 running at 2.2 MHz
Keyboard	Ergonomic design, QWERTY keyboard, repeat facility, upper & lower case numeric keypad
Interfaces	Printer interface, communications interface, cassette port, cartridge slot
BASIC	Sound and graphics commands, syntax checking, error messages, screen dump, structured features
Graphics	Teletext and viewdata compatible, max. resolution of 640 x 256, 3-d effect
Sound	Music synthesiser, 5 octaves, hi-fi output
Peripherals available	Cassette unit, floppy disk drives, hard disk drive, printers, plotter, digitiser, joystick, modem, speech synthesiser
Languages available	FORTH, PASCAL, LOGO, LISP, PROLOG, ASSEMBLER

BASIC

The computer's resident language provides commands for using the sound and graphics facilities. It checks instructions given to it to ensure that they are correct: if they are wrong it produces an error message. Screen dump reproduces the screen on the printer. Extra BASIC commands are provided to ensure that programs are written with good 'structure' —meaning that they are easy to read and correct

Sound

Individual notes or chords can be played over a range of five octaves, and the sound signal can be played through a hi-fi system

Peripherals Available

The units that can be attached to the computer include a cassette recorder, floppy disk drive and a hard disk drive. All three store programs and data. A dot matrix or a letter quality printer, a plotter and a digitiser for graphical output and input, can be used for producing words and pictures, and joysticks can be attached for games. A modem is a device for allowing computers to communicate by telephone

Graphics

The displays created by Teletext and Viewdata can be shown on the screen which has 256 rows each containing 640 dots for displaying graphics. Perspective views of three dimensional objects can be created and shown

Languages Available

These computer languages can be used instead of BASIC; each is well suited to a particular kind of application. ASSEMBLER is a kind of programming language that is more difficult to learn (than BASIC for example) but it makes programs 'run' much faster

Sinclair Spectrum

Superb colour graphics and the unique Microdrive storage system at an amazingly low price are the star features of the Spectrum — but beware the keyboard

The Sinclair Spectrum is a small personal computer with colour graphics, the ability to produce sounds and a large memory. The main reason for its success is that it is cheap — the 16K version of the Spectrum was the first colour computer to be sold for under £100. The low cost of the Spectrum has resulted in high volume sales which, in turn, have led to the existence of companies producing programs to be run on the Spectrum and extras that can be added to it.

The Spectrum offers a large amount of memory to its users, up to 48K in fact, so that it can act as a vehicle for long programs written in the machine's own BASIC computer language. These programs can be ready-written (stored on cassette, or disks), or they can be written by the Spectrum's users. The Spectrum's version of BASIC, like that of almost any other personal computer, has its own distinctive features, but it is sufficiently close to the generally accepted standard version of BASIC to be familiar to any BASIC programmer.

The facilities of the Spectrum for producing sounds and for creating graphics can both be controlled from BASIC. The Spectrum's control of sound is fairly primitive, using the appropriately named BEEP command. With some ingenuity it can be made to produce a small repertoire of sound effects. In contrast, the colour graphics capabilities are very impressive. They permit the creation of displays with features ranging from the cleverly named PAPER and INK commands for the control of background and foreground colours to commands for drawing circles and causing areas of the screen to flash on and off.

The ready-written programs that are available for the Spectrum include a tremendous variety of games. These range from the ever-popular Space Invaders and Pac-Man types to adventure games and flight simulators, and can display startling originality in their conception.

The Spectrum's capability for expansion has been satisfied to some extent by Sinclair itself with the provision of a supporting printer and storage system, the ZX Printer and ZX Microdrive. However, many other manufacturers provide add-ons for it, and these include joysticks, interfaces with which it can control or communicate with other equipment, typewriter-style keyboards, and sound and speech synthesisers.

The Sinclair Keyboard

The Spectrum has a low cost keyboard with limited movement in the keys. Sinclair opted for an improvement over the flat keyboard of the ZX81 but without incurring the extra costs of a full moving keyboard.

The Spectrum's keyboard uses a single moulded piece of rubber incorporating the keytops, which protrude through the casing. When a key is pressed it closes a contact underneath. The computer then recognises that a key has been pressed and operates the appropriate character. The means to pop the key up again is provided by the elasticity of the rubber sheet, which is stretched when the key is depressed.

This design technique has cut down the cost of the keyboard and has helped make the Spectrum the low cost computer it is

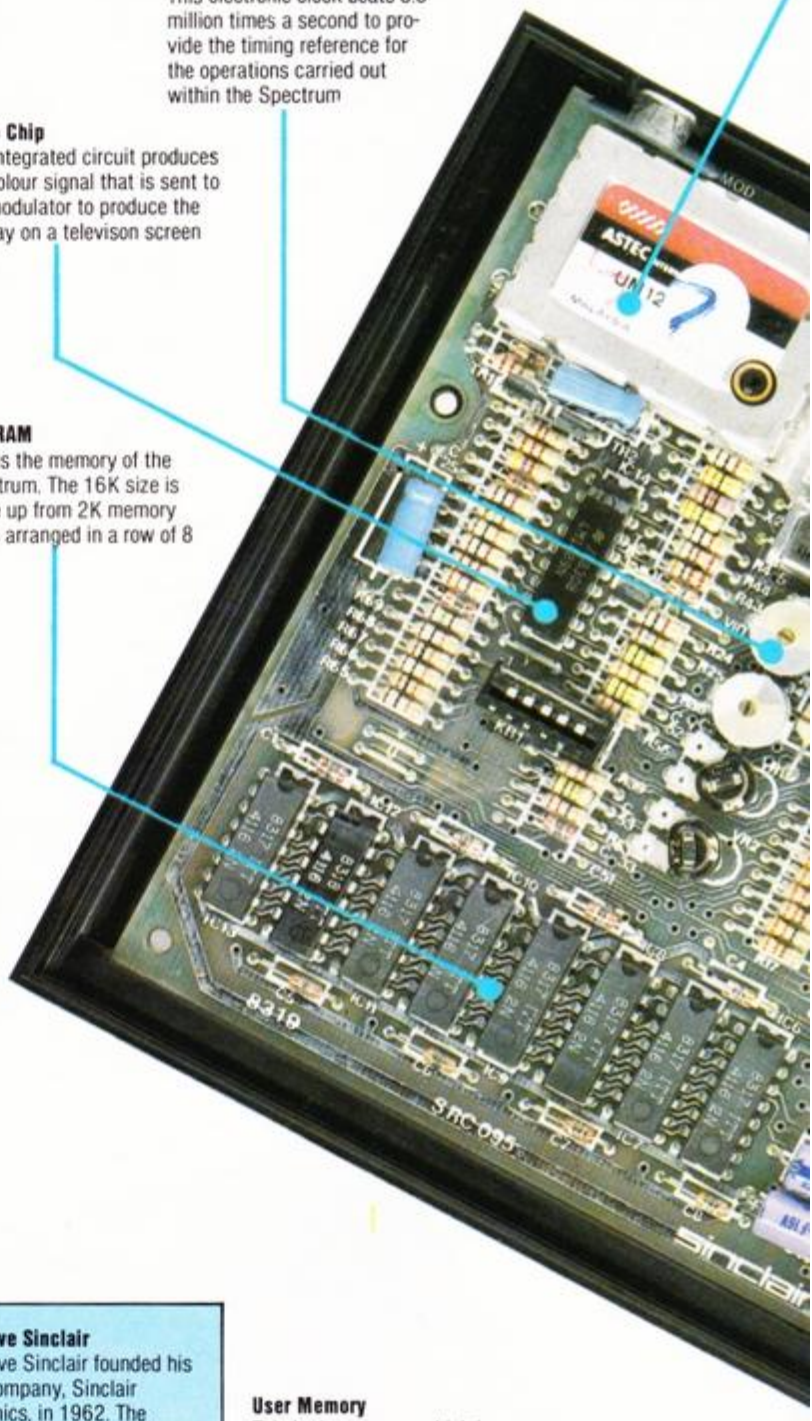
The ZX Microdrive
The ZX Microdrive connects to the Spectrum via an interface attached to the rear of the base of the machine. It provides 100 Kbytes of storage and the average time needed to access stored material is 3.5 seconds

The ZX Printer
The ZX Printer plugs directly into the Spectrum. It prints nine lines of text to the inch. The printer also produces graphics by printing the graphics characters. The contents of the screen can be copied out on the printer using the COPY command

Clock
This electronic clock beats 3.5 million times a second to provide the timing reference for the operations carried out within the Spectrum

Video Chip
The integrated circuit produces the colour signal that is sent to the modulator to produce the display on a televison screen

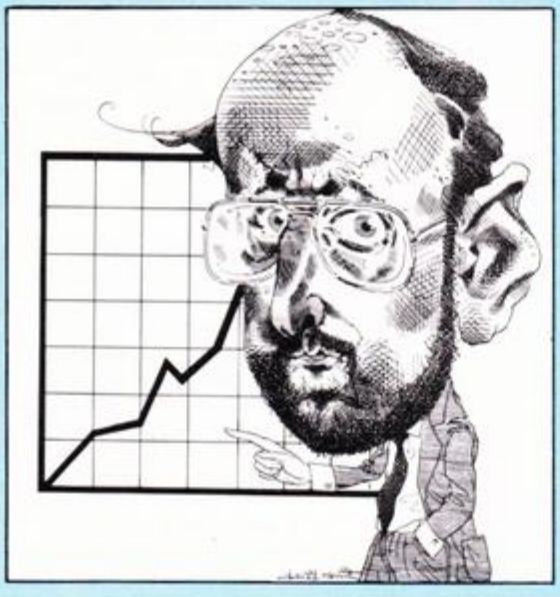
16K RAM
This is the memory of the Spectrum. The 16K size is made up from 2K memory chips arranged in a row of 8



User Memory
This is the memory which is provided with the computer for the programmer to store programs and data. The smaller of the standard allocations is 16 Kbytes, the larger is 48 Kbytes

Heatsink
This large aluminium plate dissipates any unwanted power as heat. The Spectrum will become hot after being left on for a long time, which means that this plate is doing its job

Keyboard Socket
This is where the keyboard plugs into the main compu



Sir Clive Sinclair
Sir Clive Sinclair founded his first company, Sinclair Radionics, in 1962. The introduction of the first pocket calculator, the Executive, in 1972 confirmed his flair for miniaturising and styling popular products, as well as that for selling them in huge numbers. In 1979 Sir Clive left Sinclair Radionics and founded Sinclair Research. In 1980 he developed the ZX80, followed a year later by a modified and improved version, the ZX81. These were both monochrome computers, but 1982 saw the arrival of the ZX Spectrum. In 1983 Sinclair established his own research centre in Cambridge.

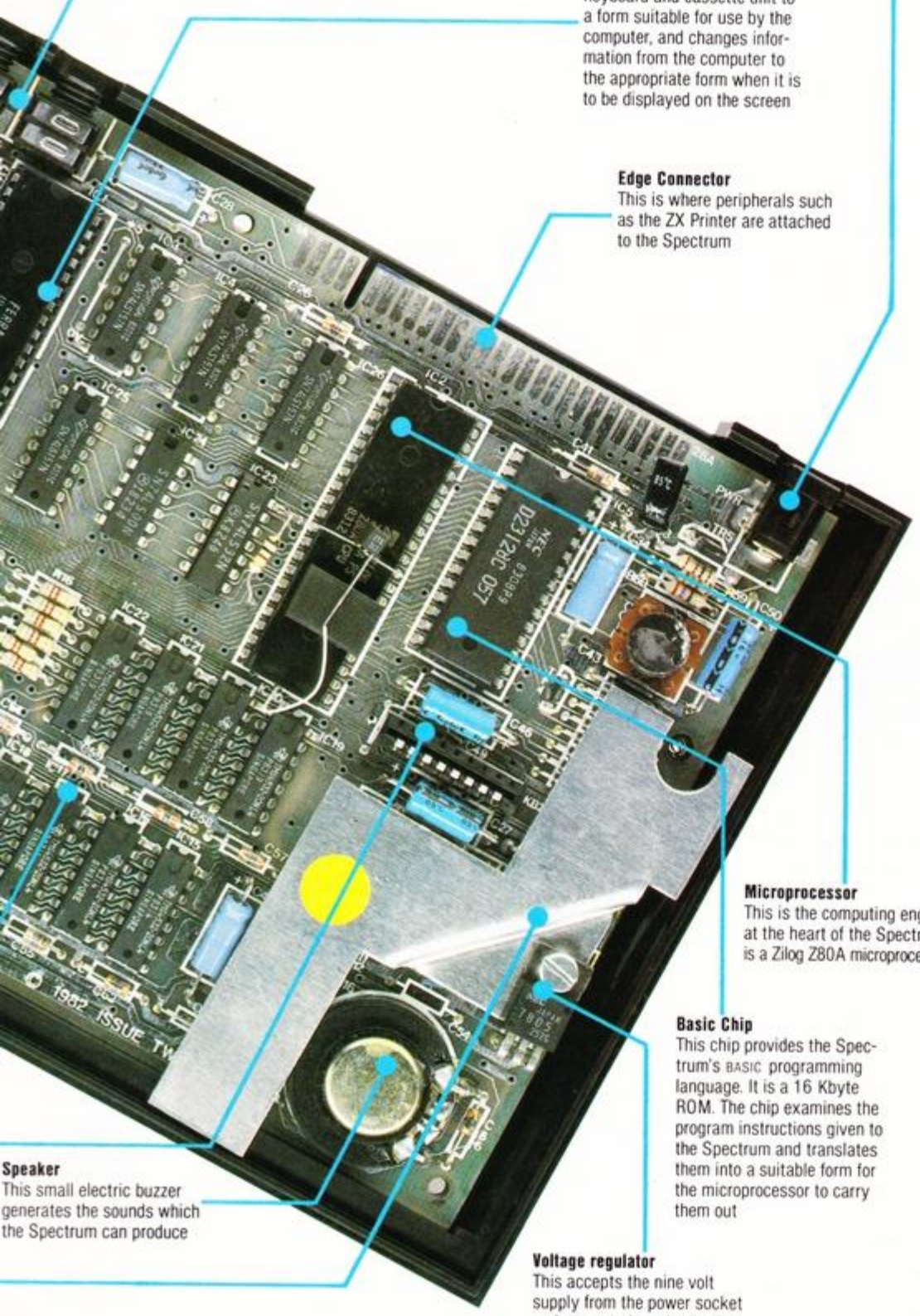
Modulator
This takes the colour signal from the video chip and converts it to a signal of the same type as that broadcast to the aerial of a television set to produce its display

Ear and Microphone Sockets
These sockets are for use with a cassette recorder to transfer information to the computer from a cassette and vice versa

Power Socket
This is where the nine volt power supply from the Spectrum's power supply unit is connected to the computer

Input/Output Chip
This converts inputs from the keyboard and cassette unit to a form suitable for use by the computer, and changes information from the computer to the appropriate form when it is to be displayed on the screen

Edge Connector
This is where peripherals such as the ZX Printer are attached to the Spectrum



Microprocessor
This is the computing engine at the heart of the Spectrum. It is a Zilog Z80A microprocessor

Basic Chip
This chip provides the Spectrum's BASIC programming language. It is a 16 Kbyte ROM. The chip examines the program instructions given to the Spectrum and translates them into a suitable form for the microprocessor to carry them out

Speaker
This small electric buzzer generates the sounds which the Spectrum can produce

Voltage regulator
This accepts the nine volt supply from the power socket and converts it to the levels needed by the various electronic components in the Spectrum

SINCLAIR SPECTRUM

PRICE
£99.95 for 16K model

SIZE
232 x 144 x 30mm

WEIGHT
552g

CPU
Z80A

CLOCK SPEED
3.5MHz

MEMORY
16 Kbytes of RAM expandable to 48 Kbytes. 16 Kbyte ROM containing BASIC

VIDEO DISPLAY
24 lines with 32 character positions, or 192 x 256 dots for high resolution graphics. Both modes have 8 colours

INTERFACES
TV connector, cassette connector (no remote control), 28-pin edge connector for connecting peripherals

LANGUAGE SUPPLIED
BASIC

OTHER LANGUAGES AVAILABLE
FORTH, PASCAL, LISP, LOGO, PROLOG

COMES WITH
Power supply unit (but no plug), aerial lead, cassette leads, demonstration cassette, 2 manuals

KEYBOARD
40 moving keys all on a single rubber sheet

DOCUMENTATION
The Spectrum comes with a thin introductory manual on setting up the machine and a more substantial manual on BASIC programming. The latter begins with a tour of the keyboard that really needs to be more detailed to describe to the beginner how the various shift keys are used. The chapters on BASIC programming demonstrate the Spectrum's capabilities with numerous examples of individual commands and a number of short programs.

A series of appendices provides a fairly complete reference guide to the Spectrum and its BASIC. The manuals are well produced and give a complete coverage, respectively, of how to set up the Spectrum and of the machine's capabilities

Simply Obeying Orders

Your computer will do exactly what you want when you 'talk' to it in the right way — and it won't make mistakes

Other Languages

BASIC is used by more people on more microcomputers than any other programming language. But BASIC is by no means the only one. Before the days of microcomputers, when most computing was done on room-sized mainframe computers, scientists and engineers used a language called FORTRAN. In the world of micros, other popular languages include PASCAL, FORTH, and LOGO

Pascal

Like BASIC, PASCAL was developed primarily as a teaching language for student programmers. It is held in high regard by teachers of programming because it encourages the writing of well thought out and elegant programs. PASCAL is usually supplied on floppy disk and tends to be expensive. A low-cost cassette version is available for the Spectrum for under £30. The PASCAL language can be used for writing large and sophisticated programs

Forth

Programs written in the FORTH language look much less like English than BASIC or PASCAL. FORTH is also more difficult to learn. It has the advantage of great power in the sense that complex programs can be written in a few lines. FORTH allows you to define your own commands, whereas in BASIC they are pre-defined

Logo

LOGO is a relatively new language becoming popular in education. It has the great advantage of being simple enough for even quite young children to learn. It can help teach programming techniques and also encourages a logical approach to program design from an early stage. LOGO uses 'turtle' graphics which allow pictures to be easily produced on the screen. A mechanical turtle (see page 34) can also be connected to the computer. Simple commands typed in on the keyboard can move the turtle and make it draw lines and shapes

It is perfectly possible for anyone to use a computer — at home or at work — without knowing anything at all about how the computer works. Starting here, THE HOME COMPUTER COURSE begins a step-by-step series that explains, from the beginning, all you need to know to be able to create your own computer programs successfully.

Many people find that after a while, the pre-packaged programs and games they've bought for their computer start to become a little boring and they wonder if they can modify them or even write their own. But a computer can do nothing by itself. It must be given a list of instructions telling it in minute detail exactly what to do and how to go about achieving it. These instructions form what is called a *program* and the art of creating them is called *programming*.

There is nothing difficult about programming. You don't even have to be good at maths, unless, of course, you want to write programs to perform mathematical tasks. All you need to begin with is to understand BASIC.

Your First Language

Most home computers are provided with a built-in computer language called BASIC. As its name implies, it is designed to enable beginners to learn the rudiments of programming quickly and easily. Like any human language, BASIC has it own grammar, vocabulary and syntax, although the vocabulary is far smaller than that of English. BASIC uses a number of short English words that are easily recognisable and simple to learn. As a general-purpose language it is suitable for both the novice and the more experienced user.

But one drawback with the language is that over the years, different computer manufacturers have tended to include their own modifications. The result is that there are a large number of variations in BASIC, particularly regarding the commands for controlling the more recently developed aspects of the machine — such as colour, graphics and sounds. Any variations of BASIC which occur in the most popular computers are shown in the 'Basic Flavours' box in each lesson.

Because of the variations in BASIC from computer to computer, it is nearly impossible to write a BASIC program of any complexity that will run on every computer. Fortunately, however, the language has a common core, which is usually the same in all machines. We'll start by concentrating on that core, and as the course progresses we will work steadily towards more complex programs.

The Initial Steps

Let's begin by writing a small program and seeing what happens. This one will show the computer apparently making a mistake. Switch on the computer and type in the program exactly as shown, including all the spaces. The <CR> at the end of each line is to remind you to hit Carriage Return. On your computer, this key may be labelled RETURN, ENTER or even ↵.

```
10 REM COMPUTERS NEVER MAKE
   MISTAKES<CR>
20 PRINT "TYPE IN A NUMBER"<CR>
30 INPUT A<CR>
40 LET A = A + 1<CR>
50 PRINT "I THINK THE NUMBER YOU TYPED
   WAS  ";<CR>
60 PRINT A<CR>
70 END<CR>
```

After you have typed it all in, type LIST<CR>. The program you just typed should reappear on the screen. LIST is an instruction to the computer to 'print' a listing of the program in memory. If the program appeared on the screen properly after typing LIST, we could try to RUN it. If you make a mistake when typing in the program, don't worry. After you have LISTed the program, simply retype any line containing a mistake. Don't forget the line number. Try typing

```
25 REM HERE IS ANOTHER 'REM' LINE<CR>
```

and then LIST the program again. To get rid of the line, type the line number alone, followed by <CR>. When you are satisfied the program has been typed correctly, you can 'run' it by typing RUN<CR>. Try this and you should see on the screen:

```
TYPE IN A NUMBER
```

Go ahead and type a number. Try 7. (Use numerals — the computer won't recognise 'seven' as 7 unless we specially program it to do so.) If you typed in 7, the screen should look like this:

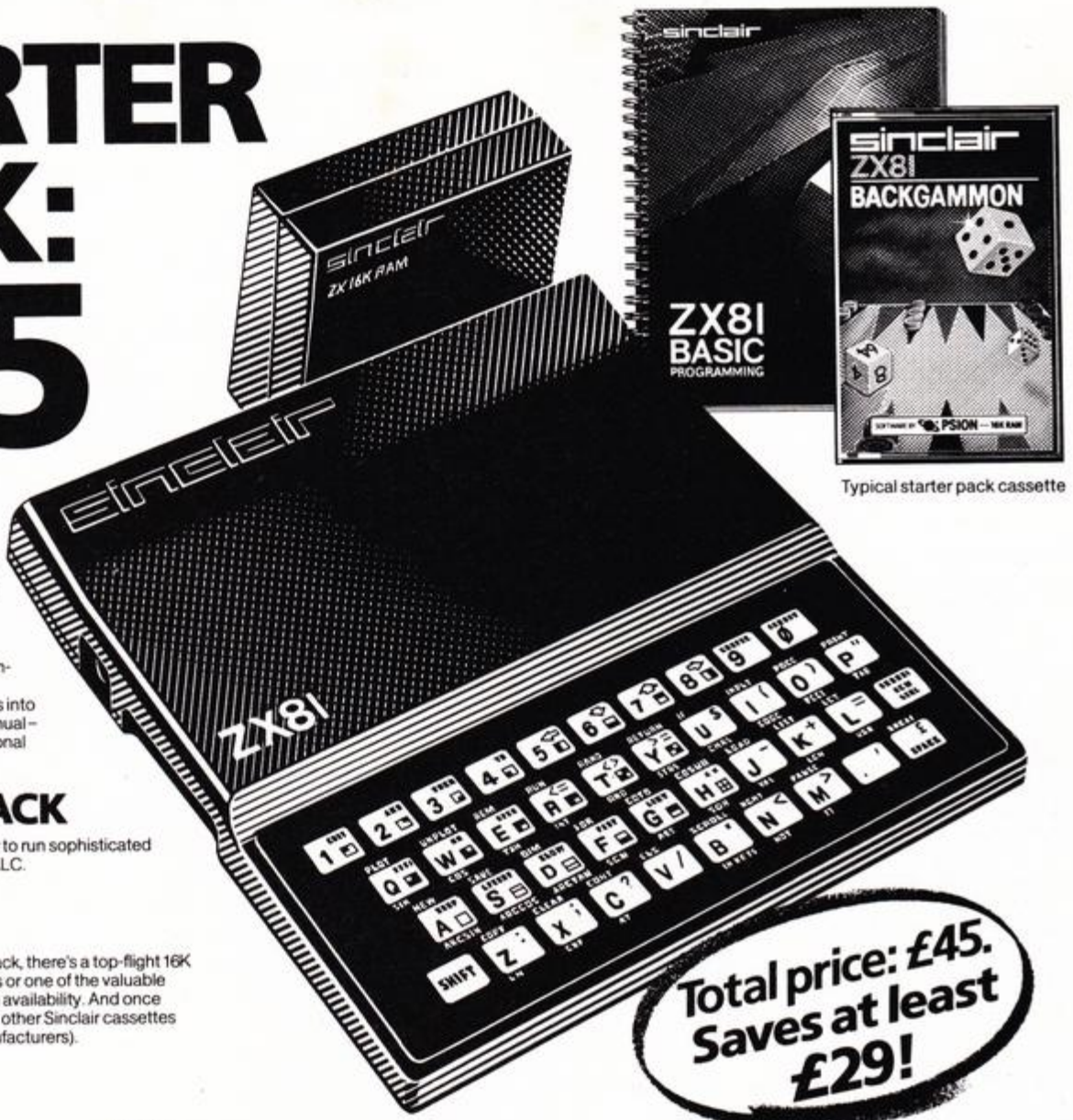
```
I THINK THE NUMBER YOU TYPED WAS 8
```

Did the computer really make a mistake, or was it simply obeying orders? If we look at the program line by line we can see what each instruction made the computer do. Here's the first line:

```
10 REM COMPUTERS NEVER MAKE MISTAKES
```

TWO SPECIAL OFFERS FROM SINCLAIR

STARTER PACK: £45



Typical starter pack cassette

Powerful passport to home computing—now at the lowest price ever!

ZX81

Sinclair ZX81 – 900,000 sold so far. Touch-sensitive keyboard... black and white display... animated graphics... just plugs into most TV sets. With 212-page BASIC manual – step-by-step guide to the world of personal computing. Normal price £39.95.

ZX 16K RAM PACK

Gives the ZX81 more power – the power to run sophisticated software like Flight Simulation or VU-CALC. Normal price £29.95.

CASSETTE

Worth £4.95 or more. In every starter pack, there's a top-flight 16K cassette – like Chess or Fantasy Games or one of the valuable education series. Actual title varies with availability. And once you own your starter pack, there are 37 other Sinclair cassettes available (plus dozens from other manufacturers).

ZX PRINTER+ FREE 5-ROLL PAPER PACK: £39.95

ZX PRINTER

Designed exclusively for use with the Sinclair ZX81 and ZX Spectrum personal computers. Printing speed: 50 characters per second. 32 characters per line, 9 lines per vertical inch. Plus graphics direct from screen. Now with a *free* pack of 5 rolls of special paper (normal price £11.95).

Look for the special packs at W. H. Smith, Boots, John Menzies, Currys and other leading Sinclair stockists.

Offers subject to availability while stocks last.

sinclair

The most popular computers in the world

MORE THAN ANY OTHER COMPUTER CAN OFFER.

FOR LESS.

Naturally, its price makes a Sinclair computer attractive.

But even more important are the other benefits that *only* Sinclair can offer.

The world's largest range of pre-written or pre-recorded home computer programs is produced for Sinclair computers.

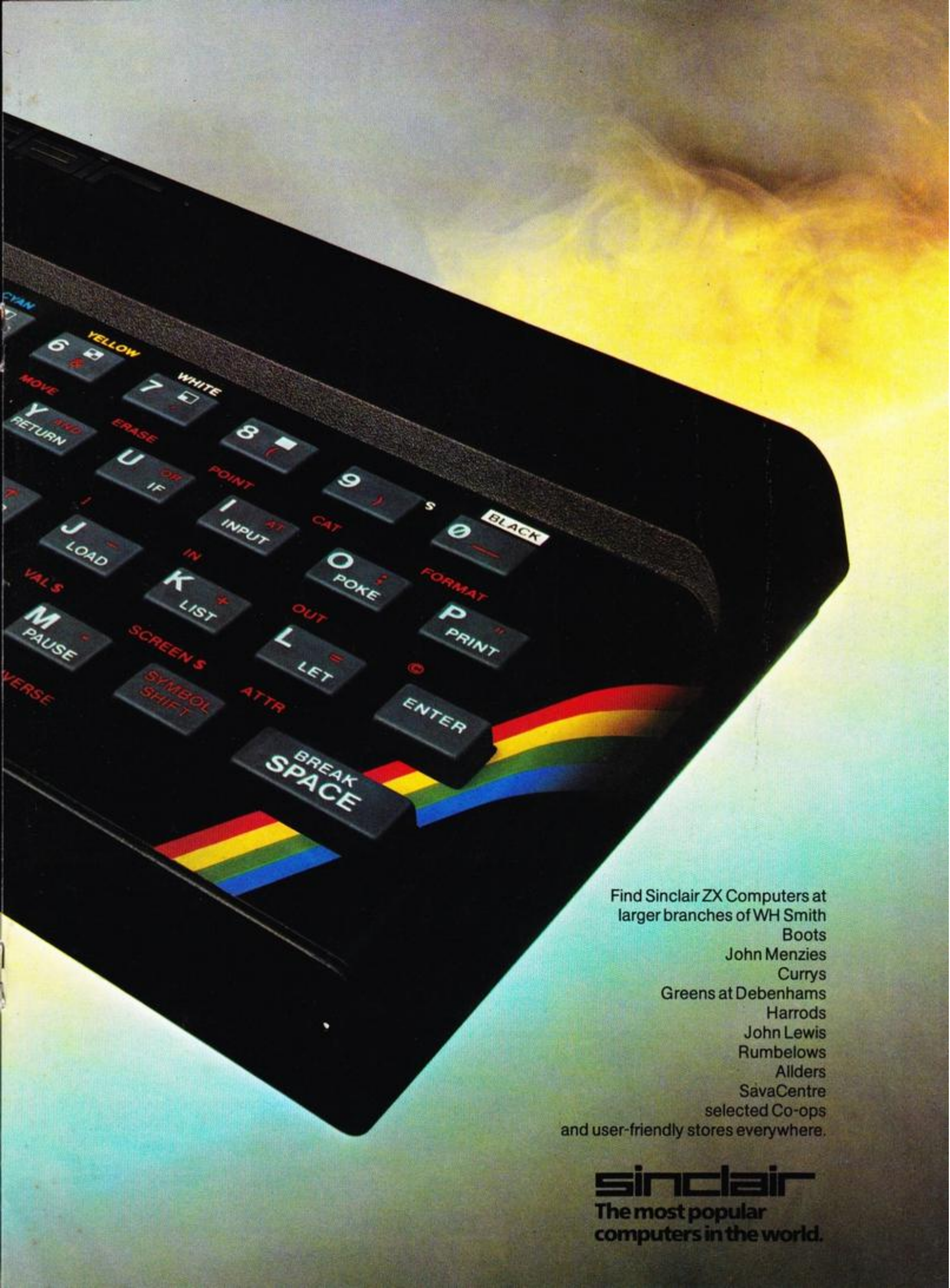
If you write your own programs, Sinclair BASIC is the world's most widely-used computer language.

Sinclair is the first manufacturer to make your child's official primary-school software available to you off the shelf.

You can buy Sinclair computers at more shops, meet more Sinclair enthusiasts, read more Sinclair magazines, join more Sinclair clubs. With Sinclair, you enter the world's biggest home-computer community.

Naturally, the unique popularity of Sinclair computers would never have built up if their performance were less than superb. Yet the most powerful computer in the world's most popular range costs under £130.

You can pay more, but you'll get less.



Find Sinclair ZX Computers at
larger branches of WH Smith
Boots
John Menzies
Currys
Greens at Debenhams
Harrods
John Lewis
Rumbelows
Allders
SavaCentre
selected Co-ops
and user-friendly stores everywhere.

sinclair

The most popular computers in the world.

Six new ways to make more of your Spectrum

Take a look at these brand-new titles. Each is an outstanding new program using the full potential of the Spectrum, for games with stunningly animated graphics, for strategies of fiendish cunning, for masterly applications of computing capability...

Cyrus-IS-Chess Based on the Cyrus Program, which won the 2nd European Microcomputer Chess Championship *and* trounced the previously unbeaten Cray Blitz machine. With 8 playing levels, cursor piece-movement, replay and 'take-back' facilities, plus two-player option. The 48K version has many additional features including an extensive library of chess openings. For 16K or 48K RAM Spectrum.

FORTH Learn a new programming language, as simple as BASIC, but with the speed of machine code. Complete with Editor and User manual. For 48K RAM Spectrum.

Horace and the Spiders Make your way with Horace to the House of Spiders, armed only with a limited supply of anti-spider-bite serum. In the house, destroy the webs before the spiders can repair them. Then destroy the spiders, before *they* destroy Horace! Undoubtedly the creepiest Horace program ever produced! For 16K or 48K RAM Spectrum.

Small Business Accounts Speeds and simplifies accounting work, produces Balance Sheets, Profit and Loss information *and* VAT returns. Complete with User manual. For 48K RAM Spectrum.

Backgammon A fast, exciting program, with traditional board display, rolling dice and doubling cube. Four skill levels. For experts – or beginners. (Rules are included – it's the quickest way to learn the game.) For 16K or 48K RAM Spectrum.

Computer Scrabble The famous board game, on-screen – with the whole board on view! Full-size letter tiles, four skill levels – the highest of which is virtually unbeatable. For 1 to 4 players. For 48K RAM Spectrum.

(Scrabble trade-mark and copyright licensed by Scrabble Schutzrechte und Handels GmbH – a J. W. Spear and Sons PLC subsidiary.)

sinclair

The most popular computers in the world.

Get your Spectrum software at most Sinclair stockists including W H Smith, Boots and John Menzies.

REM stands for REMark. Anything appearing on the same line after REM is ignored by the computer. Remarks are a handy way of reminding yourself what the computer is doing. This particular REM is just a title — it does not tell us what the program is doing. We'll see how helpful properly written REMs are later in the course. Now let's look at:

```
20 PRINT "TYPE IN A NUMBER"
```

When BASIC gets to the word PRINT, the part that follows it is 'printed' on the computer screen. Notice that the sentence is enclosed in double quote marks. One of BASIC's rules is that the characters (letters) appearing inside double quote marks after a PRINT statement will appear on the screen exactly as they were typed in. We'll see another way of using PRINT in line 60. Next comes:

```
30 INPUT A
```

We'll skip this line for now and come back to it after looking at line 40.

```
40 LET A = A + 1
```

The letter A is used here as a variable. A variable is like a labelled box that can contain either a number or some characters. Instead of having to remember what's in the box, all we have to know is what the box is called in order to reference it. It's like saying "Pass me the box labelled B" instead of "Pass me the box containing the 15mm cheesehead screws".

In this line we have a 'box' called A. This box is called a variable, because the value of what we put in it can vary. We can assign virtually any value to a variable. A value was assigned to variable A in line 30, so let's see how it was done:

```
30 INPUT A
```

Using the word INPUT is one of the ways in BASIC of assigning (giving) a specific value to a variable. When the BASIC program gets to a line starting with INPUT it waits for something to be typed in from the keyboard. INPUT A lets the computer know that we have a variable called A and that whatever is typed in at the keyboard will be assigned to that variable. Typing 7<CR> at this point puts 7 in box A, or to use computer jargon, assigns the value 7 to variable A. Now that we know what a variable is, and one of the ways of assigning a value to it, let's look at line 40 again.

```
40 LET A = A + 1
```

The name of the variable to which a value is assigned always appears on the left of the equals sign. Here we are giving a new value to A. The statement means 'LET the new value of A equal the old value plus 1.' The old value of A was 7. We have now made it 7 + 1, so the new value is 8.

```
50 PRINT "I THINK THE NUMBER YOU TYPED
   WAS ";
```

This is our print statement again. It 'prints' the character string (that is, the words or numbers you have typed) between the double quote marks. Notice the semi-colon at the end of the line. It helps to specify the positions at which things are printed on the screen. Later in the course we'll return to how the semi-colon is used in more detail. Now let's look at:

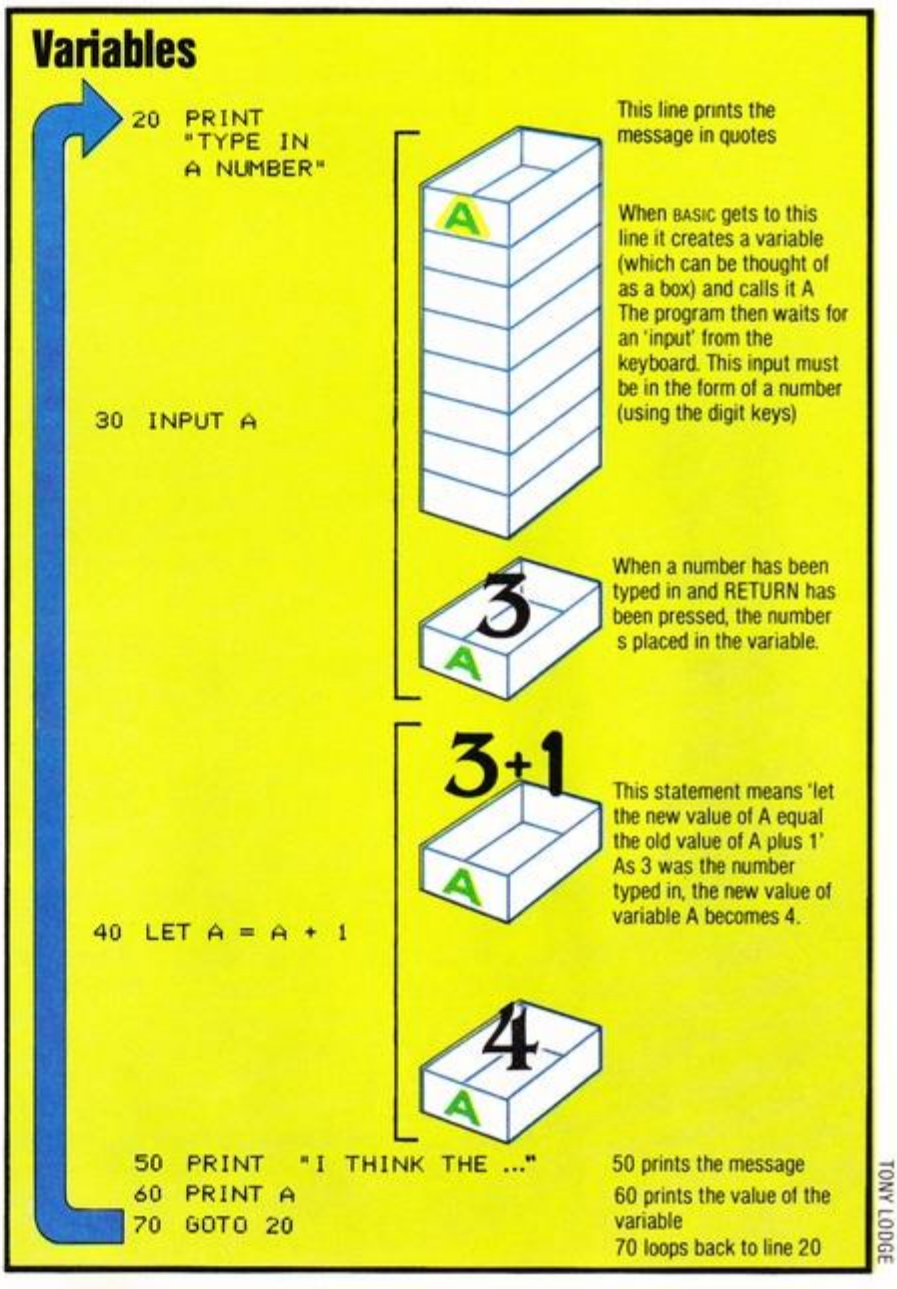
```
60 PRINT A
```

Here's another PRINT statement, but this time there are no quote marks around the A. We already know that the program will not print an actual A on the screen because we have seen that quote marks are needed to do that. Without the quotes, BASIC looks for a variable with the same label as the character after PRINT. If it finds one, it prints the value of the variable. (If it doesn't find one, it gives an error message!) This program already has a variable called A and so BASIC prints its value — what is it?

If you thought the answer was 7, remember that BASIC works through programs line by line,

The box below shows how 'variables' are used in BASIC. It also illustrates how the GOTO statement (see next page) is used to form a loop

Variables


20 PRINT "TYPE IN A NUMBER"
30 INPUT A
40 LET A = A + 1
50 PRINT "I THINK THE ..."
60 PRINT A
70 GOTO 20
A
3
A
3+1
A
4
A
This line prints the message in quotes
When BASIC gets to this line it creates a variable (which can be thought of as a box) and calls it A The program then waits for an 'input' from the keyboard. This input must be in the form of a number (using the digit keys)
When a number has been typed in and RETURN has been pressed, the number s placed in the variable.
This statement means 'let the new value of A equal the old value of A plus 1' As 3 was the number typed in, the new value of variable A becomes 4.
50 prints the message
60 prints the value of the variable
70 loops back to line 20

TONY LODGE

Basic Flavours

Only the Sinclair Spectrum uses the LET part of the instruction. On other computers this is implied, meaning it can be left off. For example, line 20 can be written as A=A + 1 instead of LET A = A + 1

This is not used on the Spectrum. The last line of the program typed in is assumed to be the end of the program

Appears on the screen as two words (GO TO) on the Spectrum, although only one key is pressed. Most other computers, with the exception of the BBC will accept the instruction typed as two words

following the order of the line numbers. By the time we got to line 60 the value of A had already been changed to 8, and that is what it will print. Finally we come to:

```
70 END
```

The END statement tells BASIC that the end of the program has been reached. Some versions of BASIC insist that all programs should finish with END while others do not (see the 'Basic Flavours' box).

Notice that when you run the program it only 'works' once. To get it to go through once more you have to type RUN<CR> again. Now we'll look at a way of getting the program to work as many times as we want by using the GOTO statement.

Using GOTO

The same program but with an extra line is given below. If you have switched off the computer to take a break, type it in. Otherwise all you need to do is to type in lines 70 and 80. These are shown in blue in the listing below.

```
10 REM COMPUTERS NEVER MAKE
   MISTAKES<CR>
20 PRINT "TYPE IN A NUMBER"<CR>
30 INPUT A<CR>
40 LET A = A + 1<CR>
50 PRINT "I THINK THE NUMBER YOU TYPED
   WAS " ; <CR>
60 PRINT A<CR>
70 GOTO 20<CR>
80 END<CR>
```

After you have typed it all in and LISTed it, see if you can figure out what will happen before you try to RUN it. Then type RUN<CR> and, as in the first version of the program, you should see:

```
TYPE IN A NUMBER
```

Type in any number (using the numeral keys) and hit RETURN. The computer will add 1 to the number and display it at the end of the message.

```
I THINK THE NUMBER YOU TYPED WAS 8
```

You will see that this is immediately followed by the TYPE IN A NUMBER message again. Entering another number and hitting return again makes the program cycle like this *ad infinitum*. The reason this happens can be found in line 70:

```
70 GOTO 20
```

When BASIC reaches a GOTO statement, instead of continuing to the next line, it GOes TO the line number specified. Here it is directed back to line 20 and the whole program is run all over again. It goes on looping back like this forever. If you want to stop the program from running you'll find there's no way of getting out of the loop. The program just goes on and on waiting for your input.

As you would expect, there are ways of writing the program so that we can get out of it if we want to, and we'll look at one of these in the next instalment of this course. Meanwhile, we still have to stop the program. If your computer has a BREAK key, it can be used to stop the program from running. Typing RUN<CR> will start the program again.

Notice that we still have the END statement at the end of the program. The way we have written this program, with the GOTO 20 statement creating an endless loop, we never do get to the end, but some versions of BASIC insist that we always use an END at the end!

If you can't find a way of stopping the program, try hitting the RESET key. That is almost certain to halt the program. Then try to LIST it again. If you get a list, you will be able to 'edit' the program in the exercises below. If you do not get a list, it means the RESET on your computer destroys the program in memory and you will then have to type the whole thing in again.

Exercises

These questions are carefully graded and are designed to be fun. Working through exercises is one of the best ways of checking that you have understood the material presented and are making genuine progress.

Before starting the exercises, try changing a few of the lines to see the effect on the way the program runs. You can't possibly do the computer any harm even if you make mistakes or hit the wrong keys. To change a line, type in the program and then check the result by LISTing it. The whole program will appear on the screen again. Type the number of the line you want to change followed by the new line. Try this:

```
10 REM COMPUTERS SOMETIMES MAKE
   MISTAKES<CR>
```

then type LIST again. Notice how the first line has been changed. If you want to get rid of everything in the line, just type the line number followed by <CR>. Try:

```
10<CR>
   LIST
```

Line 10 should have disappeared. Put line 10 back in by typing out the whole line again — not forgetting the line number!

■ Rewrite the program so that the computer really does print out the number typed in. Hint: taking out one whole line should do the trick.

■ Retype line 70 so that the program goes to line 80. LIST the program. RUN the program. Why didn't it run the same way as before?

■ Change line 60 so that the computer prints an A on the screen instead of the value of variable A.

■ Rewrite line 60 so that the computer prints the value of variable A once again. Remove line 10 (the REM line) completely. RUN the program. Does it run any differently?

■ Put in a new REMark on line 25. New lines can be added by simply typing the new number followed by the new statement. Put in a remark on line 25 to remind you what will happen next — it could be something like 'expects an input from the keyboard'. After you have typed the new line and hit <CR>, LIST the program again and check that your new remark appears in the right place.

■ Rewrite the program so that it multiplies the number you type in by 10. You'll need to change line 50 to print something like THE NUMBER YOU TYPED MULTIPLIED BY 10 IS. This time we will not want to add to the value of the old variable, we'll wants to multiply it by 10. BASIC uses the * sign to mean 'multiply'. (Don't use an X because BASIC only recognises it as a letter, not as a multiplication sign.)

We have now covered quite a lot of ground. We have seen how to write comments, which BASIC calls REMarks, how to PRINT character strings on the screen, how to PRINT the value of a variable on the screen, and how to make the program GOTO a specified line number.

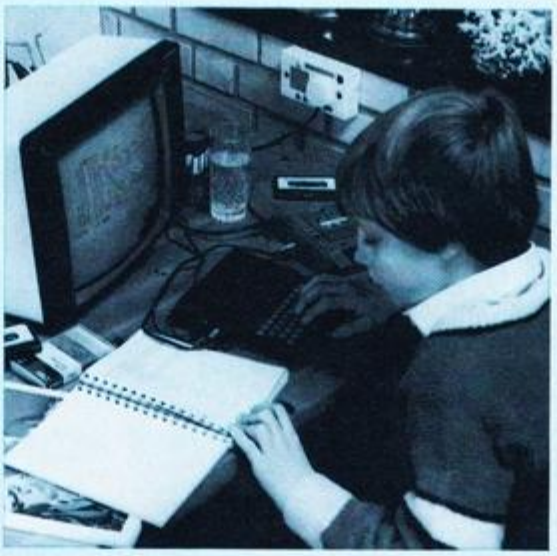
Next we'll see how to get out of a loop by using an IF-THEN statement. We'll find out how to get the program to 'perform' for us a specified number of times instead of looping forever. And we'll also see how to slow the program down to make the computer look as if it's really having to think.

And Then There Was BASIC

Today, BASIC is the world's most popular programming language. Computer languages were invented to allow the human operator to communicate more easily with the machine, and BASIC is one of the easiest to learn and use. It consists of instructions in simple English combined, where necessary, with the mathematical symbols found on a typewriter keyboard.

BASIC is a quick language to master. Within a few minutes of unpacking a microcomputer you can be writing simple programs. It was devised in 1965 at Dartmouth College, New Hampshire, with the express purpose of simplifying existing languages. The inventors were two teachers, Thomas Kurtz and John Kemeny. The universal use of BASIC has meant slight variations in the language have crept in. But the core of BASIC remains common to all manufacturers.

A program is a sequence of instructions which the computer executes to perform a specified task. The task might be to produce a monthly financial forecast, or to move a Space Invader across the television screen. The program appears as a series of numbered lines. Each line contains one instruction and the number allows the computer to obey the commands in the right order. Commands are quickly learnt and even the most complicated program uses nothing more than combinations and repetitions of the elementary commands.



BASIC has taken the mystique out of programming and made computing accessible to everyone

Most computers arrive from the manufacturers with BASIC built in. Computers can also be programmed in 'machine code' (described as 'a low' level language because it is close in structure to the logic found in the electronic circuits). BASIC is a 'high' level language as it is nearer to everyday English. There are many other high level languages devised for more technical and specialised applications, but BASIC is the best introduction to them all. It's a simple and powerful language.

World Revolution

A computer revolution is sweeping the world and changing the face of society. It's the world of tomorrow and it's happening today

The computer revolution coincided with the space race and the drive to get the first man on the moon. The billions of dollars pumped into this effort resulted in a concentration of the best scientific minds, and in breakthroughs in manufacturing techniques. The drive was to get a near-impossible job done. The spin-offs have ranged from new ceramic materials, plastics and adhesives, to the micro-miniaturisation of incredible computing power.

When 'computer' meant racks full of circuits, it also meant thousands of electronic components individually wired together. Manufacturing costs were enormous. Those same circuits can now be packed into silicon chips small enough to fit inside a case no bigger than a keyboard.

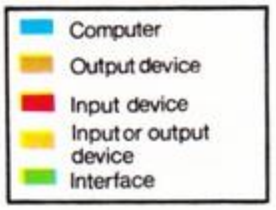
The computer chips inside a Dragon or Spectrum computer are not only small, they can be mass-produced at almost negligible cost. Silicon, the raw material from which the chips are made, is one of the world's commonest substances. Every grain of sand consists of almost nothing else.

Now that the computer has found a place in homes and offices throughout the developed world, we have the opportunity to witness first hand the start of the second industrial revolution. The original industrial revolution replaced laborious manual labour with powered machinery. The computer revolution will save skilled workers' time and replace factory workers with computer-controlled robots.

The long term impact of this revolution on our lives is far from clear. What is certain is that patterns of work and leisure will change, and fast. Robots, which are no more than mechanical extensions of computers, are replacing workers. These people are faced with the choice of learning new skills or being put out of work.

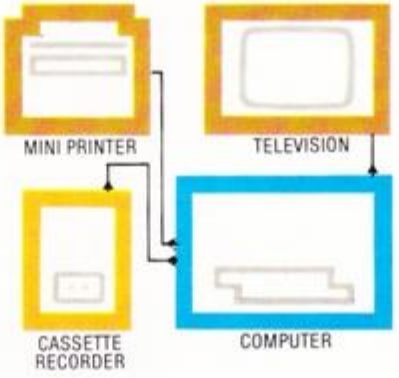
Traditional crafts such as printing and typesetting are just as vulnerable. Even teachers could be replaced using existing technology. And a visit to the family doctor could soon mean an interview with a computer terminal.

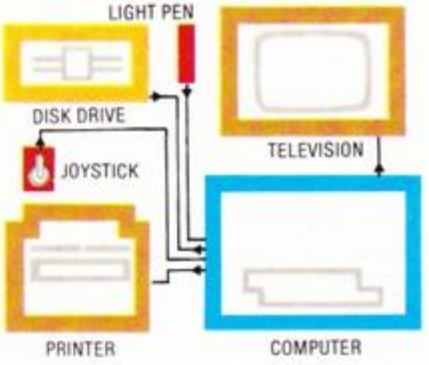
The social consequence of the first industrial revolution was to displace millions of people from the countryside and to create the industrial squalor — and material benefits — that characterise the western world. We are standing at the brink of a computer revolution just as dramatic. A revolution where the computer is the driving force of society. A revolution where understanding and using computers will be the key to survival.



Home

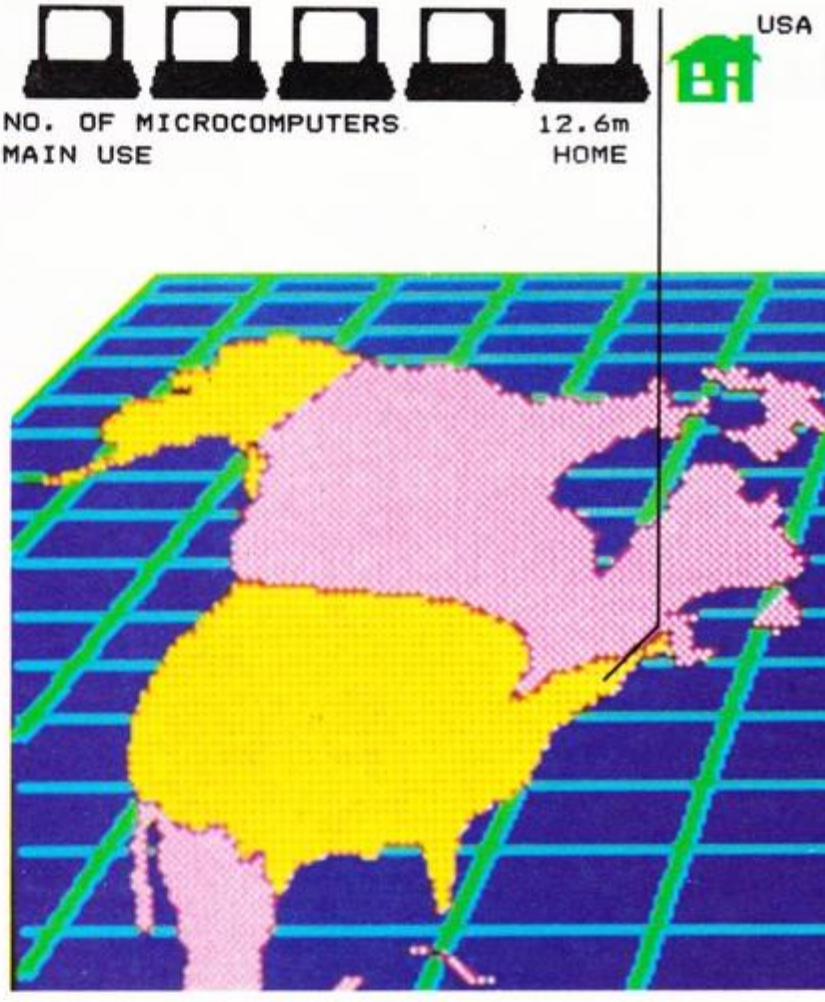
A low-cost home system can be put together for under £100. It's your passport to the future

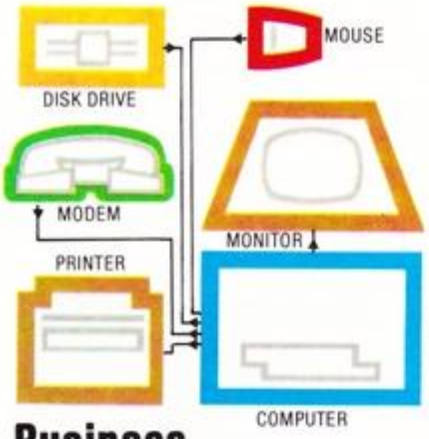




Games

A few extras open up the world of games. More than just entertaining, games teach essential principles of computing



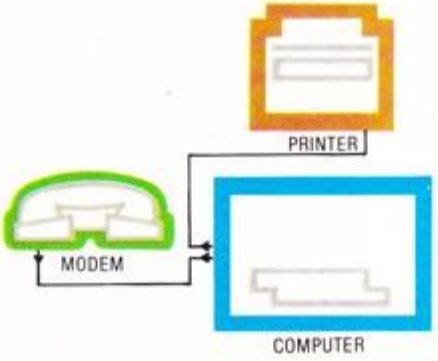


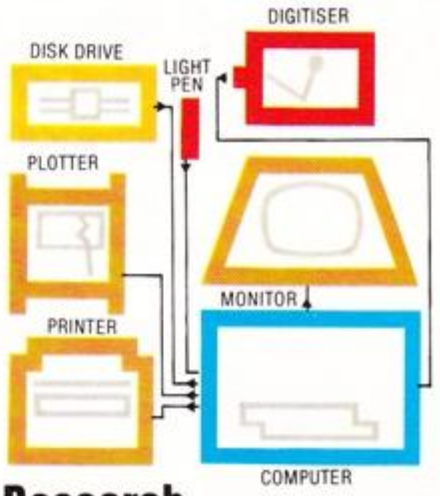
Business

Small microcomputer systems are helping — and replacing — office workers and businessmen the world over. Modems give access, via ordinary telephones, to mainframe computers and their vast data banks of vital information. Typing pools become a thing of the past as word processing takes over. Accounting and 'spreadsheet' software takes the cost out of cost accounting and makes financial decision-making a matter of minutes rather than days. Disk drives replace rooms full of filing cabinets and printers provide anything from perfectly printed letters to instant tabulations of the day's stock market quotes. Businessmen who can't type will not be left out; the 'mouse' has come to the rescue with a method of inputting commands to the computer that replaces the keyboard

Communication

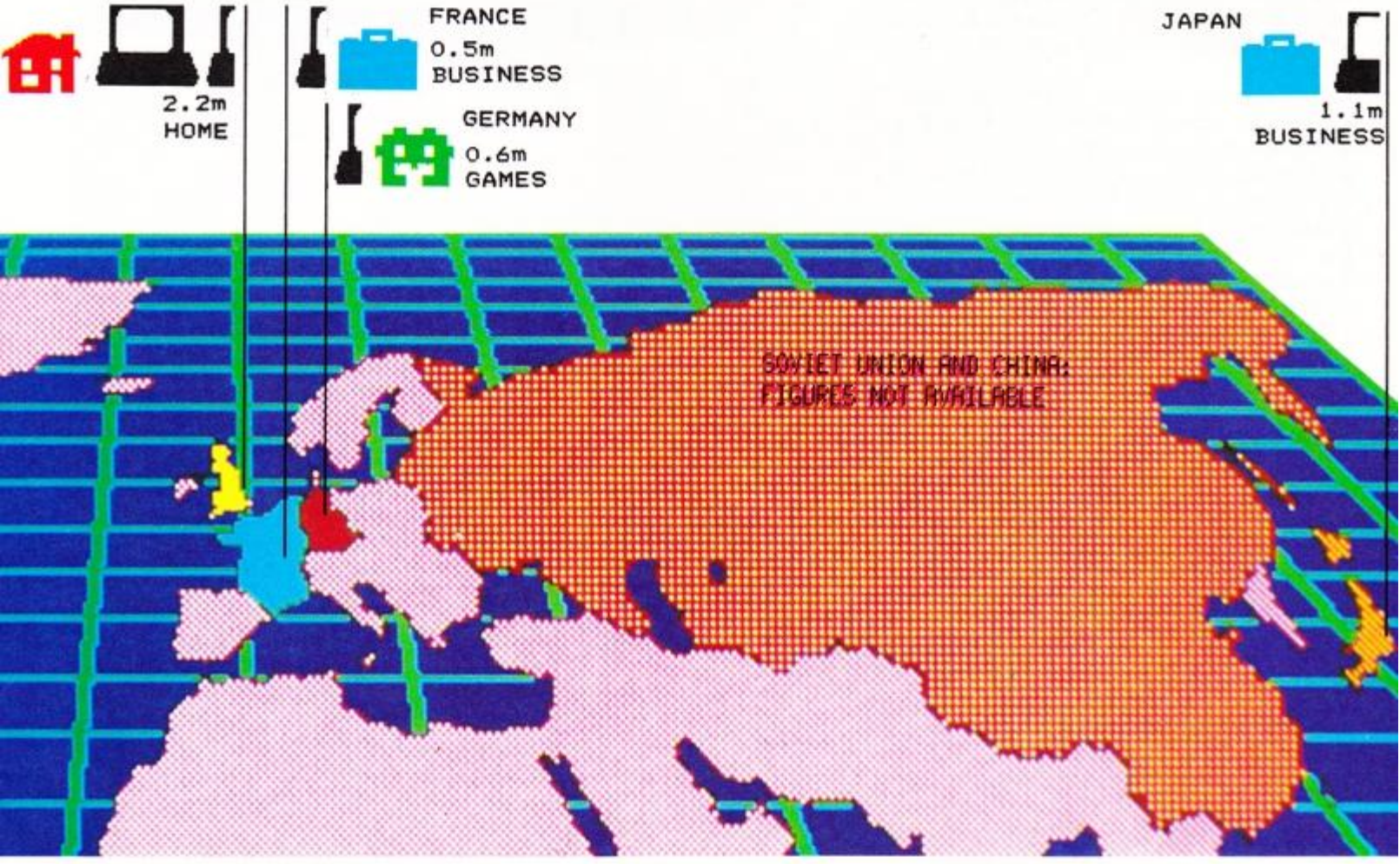
The telephone network is worldwide and many computers are now small enough to be used on the lap or held in the hand. The busy executive can type reports in a Boeing 747 on his way to New York and send them home at the speed of the electron when he arrives. The modem couples home computers to the telephone and gives instant access to virtually all the other computers in the world — including giant industrial mainframe computers. The data banks are on tap 24 hours of the day and even the journalist, filing his story from Afghanistan or South America, can get his story through in the time it takes to locate a telephone





Research

Advanced computing power is no longer the preserve of the university laboratory and giant corporation. Sophisticated systems are finding their way into schools and homes the world over. Computer Aided Design (CAD) and exploring Artificial Intelligence (AI) research is possible even with a home system. Plotters to produce technical drawings and digitisers to input complex pictures such as maps or diagrams are already affordable. Computer studies are as much a part of the school curriculum today as Latin was a few years ago. The children of today are learning to cope with the computerised world of tomorrow

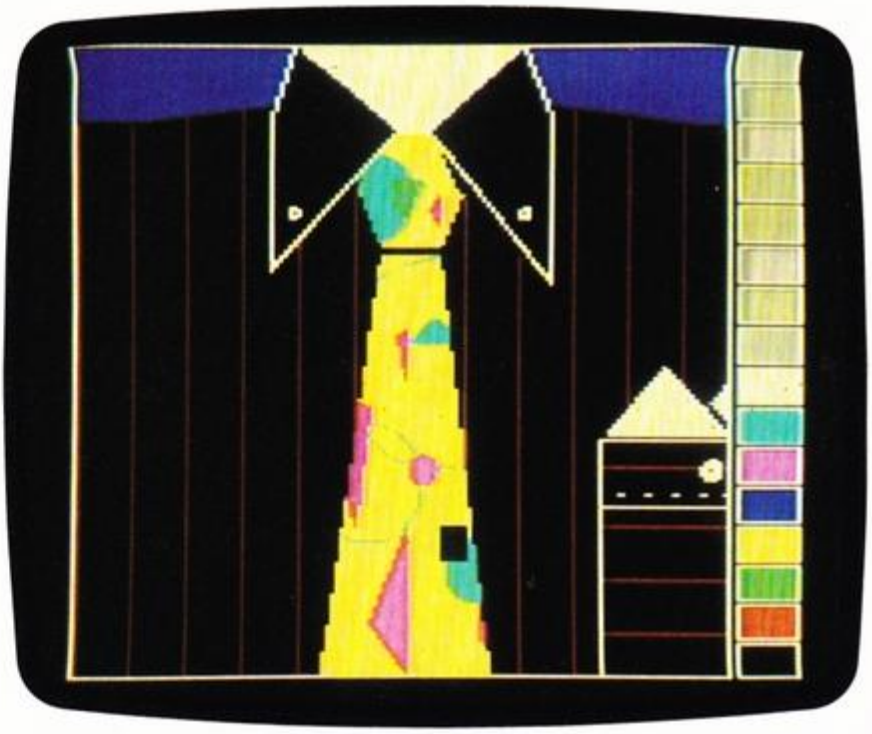


The Electronic Artist

Sophisticated or simple, computer images are all made up of thousands of tiny dots — and the computer has to remember the colour and brightness of every one

Micro Painting

Painting pictures with a microcomputer is easy with graphics software. These programs allow complex pictures to be drawn using commands typed in at the keyboard. The starting point is usually to position straight lines on the screen. Circles, triangles, squares and other pre-defined shapes can be added. Colour can be used both for lines and for whole areas of the screen. Areas can be painted by selecting a colour, either from a palette displayed on the screen or through the keyboard. The area enclosed by its boundary lines is automatically coloured. In some programs a 'brush' can be moved across the screen like a cursor. The large photograph shows the result of this 'building' process. (The software was 'Graphkey' on a BBC Model B)



You're skimming the rooftops in an aeroplane, one engine dead, a skyscraper looming in front of you — and the runway you're trying desperately to land on already has a plane burning in the middle of it. There it is in front of you on the television screen. That is just one form of computer graphics.

With educational programs, such as those designed to teach spelling or arithmetic skills to children, it is clear that unless a program produces an interesting visual display, the attention of young people will be unlikely to be held for long.

Business users of microcomputers will probably be dealing primarily with numbers representing amounts of money that have been received or spent, the stock held of particular items and so on. But this kind of information is far more readily understood and interpreted when shown pictorially The computer's ability to 're-draw' a picture rapidly, so as to incorporate new information, different alternatives, and so on, and to produce 'hard' (printed) copies when required is also of immense value in many business applications.

The Computer's Canvas

How does a computer create pictures? To answer this question let us first look at the 'canvas' on which the computer works. A microcomputer produces images on its display screen by illuminating or 'turning on' one or several dots at any of a number of positions on the screen. These dots are arranged in rows across the screen and in columns down it, so the location of any dot can be given by its position in row and column. Specific images (both pictures and print) are produced by the illumination of certain dots while the others remain unlit. This is true not only for monochrome screens, but for colour displays too. Here pictures are produced as a result of dots being made to assume the appropriate colours.

To display a single letter or number, the computer uses a rectangular array of dots. This array is known as a 'dot matrix'. On a typical microcomputer this is likely to consist of a block of eight rows each containing eight dots.

The number of dots on the screen is not the same for every microcomputer, but a fairly typical grid would consist of 192 rows each containing 256 dots — in other words 192 rows and 256 columns.

It is clear that the more dots on the microcomputer's display screen, the more detailed the images can become. The degree of screen detail that can be shown when displaying graphics is known as *resolution*. A computer that can display 192 rows each with 256 dots is said to have a resolution of 256 x 192. The higher the resolution — that is, the more dots that can be placed on the screen — the less 'grainy' the image.

All microcomputers have a maximum density of dots that can be displayed on their graphics screens, but some computers can also be programmed in ways that use less dense arrays of dots. For example, the BBC Microcomputer has a maximum resolution of 640 x 256; that is, it has a maximum dot density given by 256 rows each containing 640 dots. However, it can also be programmed using only 320 of these columns, a resolution of 320 x 256, or even lower resolutions if required. On a machine that is capable of providing this kind of variation, the resolution must be set at the beginning of the part of the program that produces the graphics.

Lines and curves produced by a dot-based computer graphics system are not actually continuous, as they would be if they were drawn with a pen. Instead they are paths of illuminated dots that merge more or less closely depending on the resolution of the system. One with low resolution will be able to produce only lumpy curves, and a straight line will not necessarily be perfectly straight either, since the dot positions on the screen lie in straight lines only in certain directions — such as along rows and columns, and across the diagonals of the array of dots. To plot a straight line the graphics system must illuminate the dots that lie closest to the path of the chosen line. The result can be a 'staircase' effect. Again, the higher the resolution of the system, the less noticeable such 'staircases' will be.

Because graphics are shown on a television screen, in order to give continuity of display the image must be continually 'refreshed' or 'redrawn' — otherwise it would appear only for a moment and then fade. For this reason, the image must be represented in some way in the computer, so that it can be referred to when necessary. The representation of the image is actually stored in a special area of the computer's memory known as the *screen memory*. With a monochrome display, each dot on the screen corresponds to a bit in the screen memory. An image is represented by setting the bits that correspond to illuminated dots to 1 and leaving the bits corresponding to unlit dots at 0. Thus, if a computer can maintain a monochrome display with a resolution of 256 x 192 it must have a screen memory of 256 x 192 or 49,152 bits — which is 6 kilobytes, since 1 kilobyte equals 8,192 bits.

With colour graphics more memory is required. Two bits can be used to represent four different colours, perhaps as follows:

bit 1	bit 2	colour
0	0	white
0	1	red
1	0	blue
1	1	black

To represent any four-colour image, it is necessary to assign two bits to each dot on the screen. In the same way, for eight colours three bits in the screen memory are assigned to each dot on the screen, while for 16-colour displays there are four bits per dot, and so on. So, a microcomputer that can display colours with a resolution of 160 x 256, must have a screen memory of 160 x 256 x 4 which is equivalent to 160 kilobits (20 kilobytes).

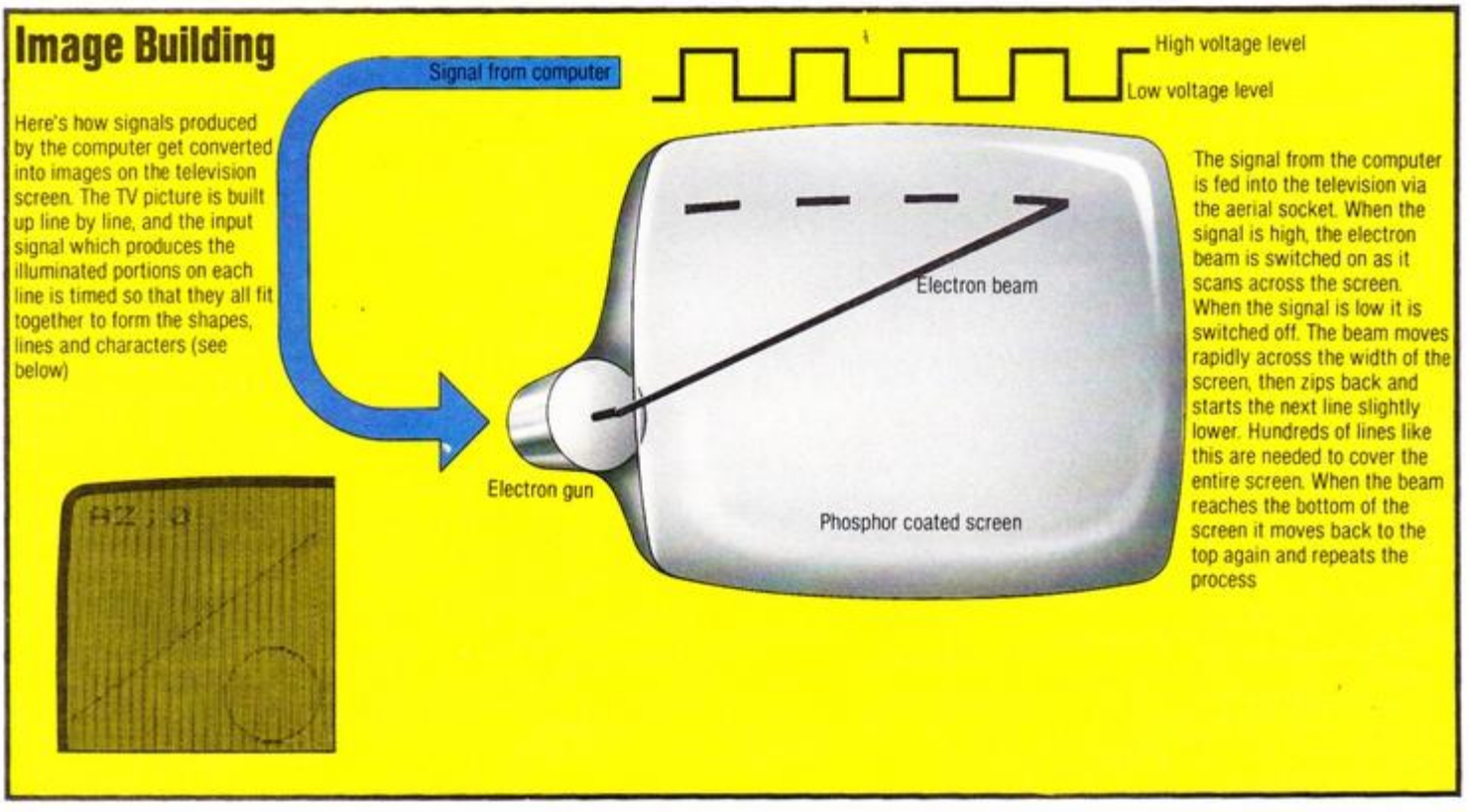
The need for screen memory explains in part why some computers are designed to work at different resolutions. Without a sufficient screen memory the computer cannot store and therefore cannot display high resolution pictures

Creating Graphics

Building images on the home computer can be easy. Most models have special commands in BASIC to help 'draw' or define objects on the screen ranging from invader-style aliens to whole pictures. Special software is also available to create animated pictures on the screen.

Image Building

Here's how signals produced by the computer get converted into images on the television screen. The TV picture is built up line by line, and the input signal which produces the illuminated portions on each line is timed so that they all fit together to form the shapes, lines and characters (see below)



The signal from the computer is fed into the television via the aerial socket. When the signal is high, the electron beam is switched on as it scans across the screen. When the signal is low it is switched off. The beam moves rapidly across the width of the screen, then zips back and starts the next line slightly lower. Hundreds of lines like this are needed to cover the entire screen. When the beam reaches the bottom of the screen it moves back to the top again and repeats the process

Bits And Bytes

The computer understands nothing but numbers — but they're numbers with an unfamiliar look about them

The words 'bits' and 'bytes' are used whenever computers are written about. They are terms that describe the way computers store and use numbers.

They do this quite differently from the way that people do. We represent numbers with 10 different symbols (O through to 9) and manipulate them in multiples of 10. (This is known as a 'base' of 10). Computers, on the other hand, and for all their mathematical wizardry, use only two numbers — zero and one. Bits and bytes are ways of representing combinations of these two numbers.

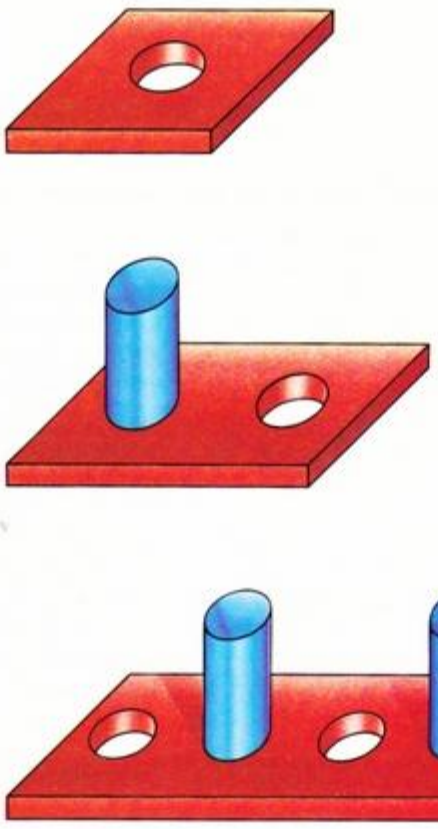
A bit is the smallest piece of information a computer can handle. It is the computer's way of representing the two numbers zero and one. A group of eight bits is called a byte; a byte allows the computer to represent quite large numbers.

First, let's look at bits, what they are, and why they are called 'bits'. Computers are electronic devices and, consequently, everything they do is done ultimately with electrical signals. A single electrical signal can either be 'on' or it can be 'off'; it is this principle that allows ordinary electrical signals to represent numbers.

The illustration shows a piece of wood with a hole in it, which can be filled with a peg. Even though it is a single hole, it can represent two numbers and is an excellent analogy for the way a computer works. Either the hole has no peg in it — in which case it represents a zero — or it does have a peg in it, and this represents a one. A single board can thus symbolise a zero or a one.

In a computer, the same effect is achieved with an electrical signal: when it is off, it represents a zero; when it is on, it represents a one. A single wire, or a one-hole board, can therefore be used to represent two states: no peg or with a peg; absent or present; off or on; 0 or 1.

This smallest piece of information is called a bit. The word itself suggests its small size and represents two possible states. The word is derived from **BI**nary digi**T**. Thought of in another way, a bit can



Bits And Bytes

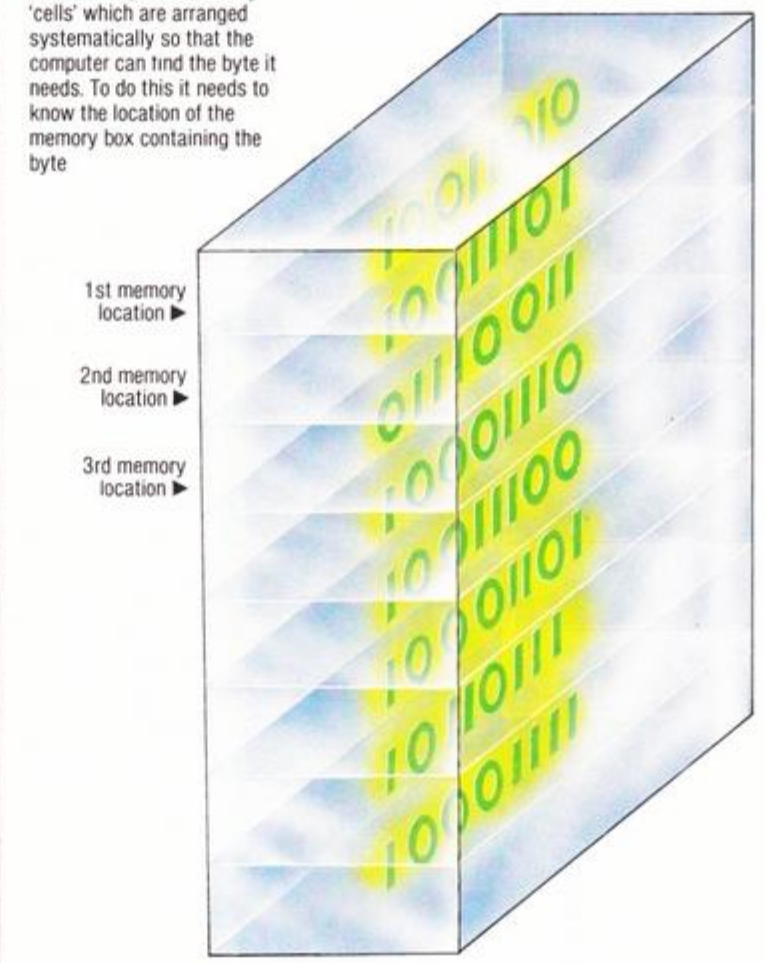
A bit (binary digit) is the smallest unit of mathematical information a computer can handle. Here, a board with just a single hole can represent either zero or one depending on whether it has a peg in it or not. A computer does the same thing with an electrical signal that is either off (to represent 0) or on (to represent 1). If the board has two holes, there are four possible combinations of holes and pegs. If eight holes are used there are 256 different combinations and these are all shown in the table. Computers use groups of eight bits and such a group is called a byte. Each byte can represent a number ranging from 0 to 255

TONY LODGE

0	00000000	128	10000000
1	00000001	129	10000001
2	00000010	130	10000010
3	00000011	131	10000011
4	00000100	132	10000100
5	00000101	133	10000101
6	00000110	134	10000110
7	00000111	135	10000111
8	00001000	136	10001000
9	00001001	137	10001001
10	00001010	138	10001010
11	00001011	139	10001011
12	00001100	140	10001100
13	00001101	141	10001101
14	00001110	142	10001110
15	00001111	143	10001111
16	00010000	144	10010000
17	00010001	145	10010001
18	00010010	146	10010010
19	00010011	147	10010011
20	00010100	148	10010100
21	00010101	149	10010101
22	00010110	150	10010110
23	00010111	151	10010111
24	00011000	152	10011000
25	00011001	153	10011001
26	00011010	154	10011010
27	00011011	155	10011011
28	00011100	156	10011100
29	00011101	157	10011101
30	00011110	158	10011110
31	00011111	159	10011111
32	00100000	160	10100000
33	00100001	161	10100001
34	00100010	162	10100010
35	00100011	163	10100011
36	00100100	164	10100100
37	00100101	165	10100101
38	00100110	166	10100110
39	00100111	167	10100111
40	00101000	168	10101000
41	00101001	169	10101001
42	00101010	170	10101010
43	00101011	171	10101011
44	00101100	172	10101100
45	00101101	173	10101101
46	00101110	174	10101110
47	00101111	175	10101111
48	00110000	176	10110000
49	00110001	177	10110001
50	00110010	178	10110010
51	00110011	179	10110011
52	00110100	180	10110100
53	00110101	181	10110101
54	00110110	182	10110110
55	00110111	183	10110111
56	00111000	184	10111000
57	00111001	185	10111001
58	00111010	186	10111010
59	00111011	187	10111011
60	00111100	188	10111100
61	00111101	189	10111101
62	00111110	190	10111110
63	00111111	191	10111111
64	01000000	192	11000000
65	01000001	193	11000001
66	01000010	194	11000010
67	01000011	195	11000011
68	01000100	196	11000100
69	01000101	197	11000101
70	01000110	198	11000110
71	01000111	199	11000111
72	01001000	200	11001000
73	01001001	201	11001001
74	01001010	202	11001010
75	01001011	203	11001011
76	01001100	204	11001100
77	01001101	205	11001101
78	01001110	206	11001110
79	01001111	207	11001111
80	01010000	208	11010000
81	01010001	209	11010001
82	01010010	210	11010010
83	01010011	211	11010011
84	01010100	212	11010100
85	01010101	213	11010101
86	01010110	214	11010110
87	01010111	215	11010111
88	01011000	216	11011000
89	01011001	217	11011001
90	01011010	218	11011010
91	01011011	219	11011011
92	01011100	220	11011100
93	01011101	221	11011101
94	01011110	222	11011110
95	01011111	223	11011111
96	01100000	224	11100000
97	01100001	225	11100001
98	01100010	226	11100010
99	01100011	227	11100011
100	01100100	228	11100100
101	01100101	229	11100101
102	01100110	230	11100110
103	01100111	231	11100111
104	01101000	232	11101000
105	01101001	233	11101001
106	01101010	234	11101010
107	01101011	235	11101011
108	01101100	236	11101100
109	01101101	237	11101101
110	01101110	238	11101110
111	01101111	239	11101111
112	01110000	240	11110000
113	01110001	241	11110001
114	01110010	242	11110010
115	01110011	243	11110011
116	01110100	244	11110100
117	01110101	245	11110101
118	01110110	246	11110110
119	01110111	247	11110111
120	01111000	248	11111000
121	01111001	249	11111001
122	01111010	250	11111010
123	01111011	251	11111011
124	01111100	252	11111100
125	01111101	253	11111101
126	01111110	254	11111110
127	01111111	255	11111111

Bytes In Memory

Bytes are groups of eight binary digits (bits). Each byte is used by the computer to store numbers which can range from 0 to 255. Each byte is stored in separate memory 'cells' which are arranged systematically so that the computer can find the byte it needs. To do this it needs to know the location of the memory box containing the byte



TONY LODGE

count, but only from zero to one.

A board with two holes can show four different states, or count from 0 to 3. Both holes can be empty; the right hole can have a peg; the left hole can have a peg or both holes can have pegs. The bottom of the picture shows a board with eight holes. There are 256 possible permutations of pegs and holes and these are shown in the table using ones to represent pegs and zeros to represent holes.

Such a group of eight binary digits (bits) is called a byte. A single byte can therefore represent 256 different states (it can count from 0 to 255).

When we say a computer 'stores' a byte, we mean that a number (ranging from 0 to 255) is kept in the computer's memory, to be used when required. Each byte has its own 'box' and these 'boxes' one arranged in sequence (the picture above shows them stacked one on top of the other). When the computer needs to retrieve a number from a memory box, it simply needs to know in which box the byte is stored.

All the numbers from 0 to 255 can be represented using unique combinations of ones and zeros (table on left). Bits are stored and used by computers in groups of eight. Eight bits together are called a byte

Oric-1

This low-priced British computer has impressive colour graphics and a wide variety of sound effects

The Oric-1, a small British-made computer, competes with the Sinclair Spectrum in both cost and capabilities. It comes in a neat grey plastic housing with the keyboard tilted at a comfortable angle for typing. The keyboard has individual moving keys and touch-typing is just possible.

Two versions are available; the more expensive offering 48 Kbytes, enough to store substantial programs.

The Oric has the usual connections for television, cassette and other units. It can be linked to a printer and has another socket for plugging in extra memory, program cartridges and a modem.

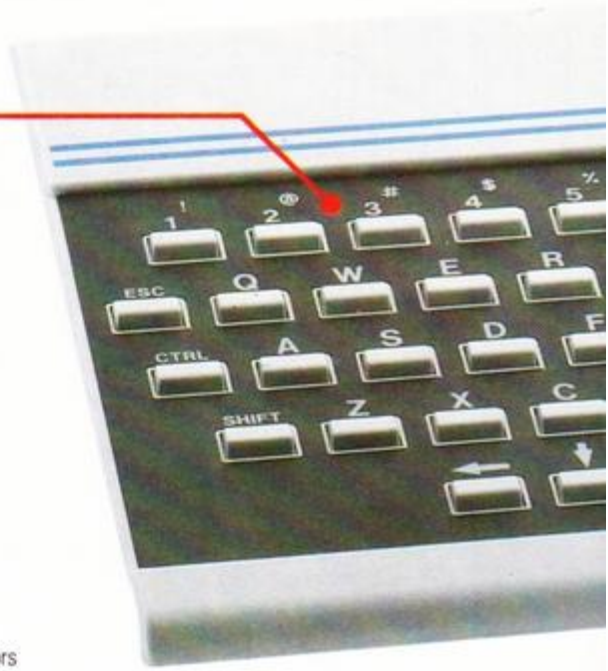
The modem is a particularly exciting add-on. It allows the Oric to communicate with other computers by telephone. The modem can give the Oric access to the Prestel computer, enabling the user to send and receive 'electronic mail'.

The Oric has BASIC built in and it is also possible to work with other languages. The 48K version is supplied with FORTH as well as BASIC.

Colour graphics and sound can be generated using the Oric's resident BASIC. Eight colours can be displayed and characters of any shape may be created and stored. PAPER and INK commands allow one to change the colour of any of these 'defined' characters and the colour of the background against which they are set.

The Oric's sound is as impressive as its graphics. Special commands permit a wide variety of sounds and music to be produced. Musical notes and chords can extend over six octaves.

The Oric is an inexpensive micro of great versatility. Its potential for expansion makes it particularly attractive, but its ability to communicate via the telephone adds a special excitement.



Keyboard
The Oric's keyboard has 57 moving keys. The letter and numeral keys are arranged in the standard QWERTY layout. The ESCAPE and CONTROL keys are on the left, and DELETE and RETURN on the right. The bottom row of the keyboard contains the space bar and cursor control keys. Because the keys are arranged in the same way as on a typewriter, and move individually, it is possible with experience to type at considerable speed on this keyboard

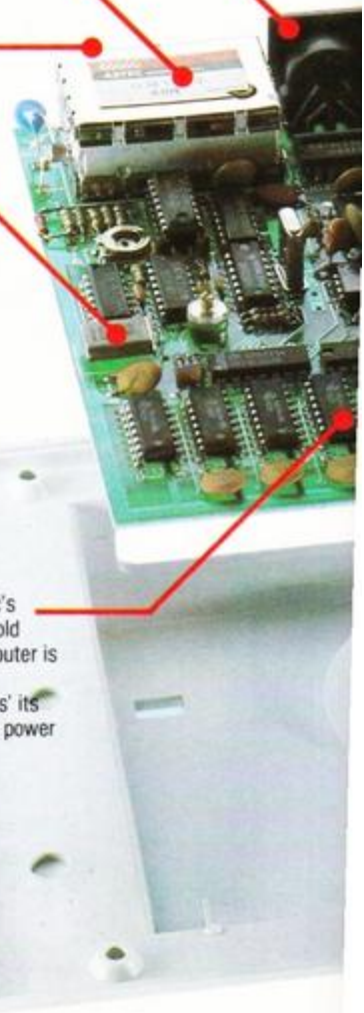
RGB Socket
This allows the Oric to be connected to colour monitors using separate red, green and blue (RGB) signals for better quality screen displays

RF Modulator
The video signal produced by the computer cannot be fed directly into a television aerial socket. This circuit converts the signal into a form suitable for an ordinary television

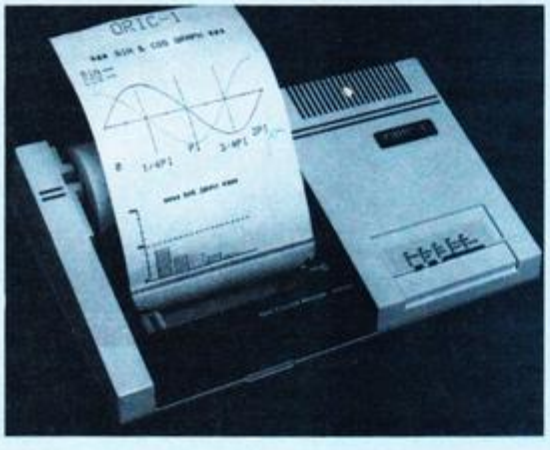
Cassette Socket
The cassette recorder connected to the com through this socket

Television Socket
The Oric is connected to a television set through this socket

Clock
The electronic clock beats one million times a second to provide the timing and synchronising of all the operations carried out by the Oric



RAM
These chips are the Oric's main memory, used to hold software while the computer is running. RAM (Random Access Memory) 'forgets' its contents as soon as the power is switched off



The Printer
The colour printer is styled to match the appearance of the Oric-1. It can print text and plot graphics in the four colours of red, green, black and blue.

The printer uses four small ball-point pens, with one for each colour, and they produce its plots by writing on an 11cm wide roll of paper. When writing text, it can produce characters in any of 15 different sizes and at four different angles.

With this degree of flexibility the printer can do far more than simply produce permanent copies of program listings

Printer Interface
This socket provides a standard connection through which a printer can be attached to the computer

Expansion Port
This socket allows a wide range of devices to be connected to the Oric, including program cartridges and communications equipment

Power Socket
The power supply from the Oric's separate transformer is connected here

Heatsink
Electronic circuits can easily be damaged by high temperatures. This safely dissipates any excess heat generated

ROM
ROM memory (Read Only Memory) stores software permanently. The main function of these ROM chips is to store the BASIC programming language

Microprocessor
This chip is the workhorse that provides the computer with the ability to carry out computations. It is a 6502A microprocessor

Input/Output
This chip converts signals entering the Oric into a form usable by its CPU. Outgoing data is converted to the type of signal needed by printers and other devices

Sound Chip
A purpose designed integrated circuit enabling the computer to produce advanced sound effects

Video Chip
The Oric is known for its good colour. The chip generates a signal containing colour data. This is sent on via the modulator, to the television screen

Speaker
The sound quality is startling. The Oric uses its own loudspeaker rather than that of the television

Oric-1

PRICE
£129.95 for 16K model with starter pack

SIZE
280 x 178 x 150mm

WEIGHT
848g

CPU
6502A

CLOCK SPEED
1MHz

MEMORY
16 Kbytes of RAM expandable to 48 Kbytes.
16 Kbytes ROM containing BASIC

VIDEO DISPLAY
4 modes: Text mode with 28 rows of 40 characters and 2 low-resolution graphics modes on the same grid; high-resolution mode with 200 x 240 dots

INTERFACES
TV connector, cassette and hi-fi connector, Centronics printer interface, RGB video, expansion connector

LANGUAGE SUPPLIED
BASIC

OTHER LANGUAGES AVAILABLE
FORTH

COMES WITH
Power supply unit with integral plug, aerial lead, cassette leads, game on cassette, manual

KEYBOARD
57 individual moving keys, including a space bar

DOCUMENTATION
The Oric comes with a BASIC programming manual that gives a racily written introduction to the machine and its BASIC. The chapters on BASIC programming deal with the handling of numbers and words, and show how to use the Oric's colour graphics and sound facilities to good effect.

There are many short programs that are instructive on an elementary level. Appendices are a necessary way of organising the detailed information, but there is no index. All in all, the documentation is entirely adequate for the beginner

Getting In Touch

Keyboards at first seem much alike. But some are distinctly better than others, and work in quite different ways

A computer's keyboard is an important part of the system. It is, after all, the way you communicate with the computer. The keyboard must be given as much consideration as the memory capacity, or the quality of the graphics.

Microcomputers have inherited the typewriter's Qwerty-style keyboard — so-called because the first six letters on the top row of keys spell QWERTY. In the early part of this century, the individual characters were positioned on the keys in such a way as to slow typists down so that they wouldn't wreck the flimsy mechanisms!

By the early 1950's when computers first came into commercial use, the QWERTY layout was the standard system for typists and became the standard entry device for computers as well. Today's microcomputer owner is stuck with the QWERTY system which is fine for trained typists, but sometimes difficult for the newcomer to master.

When computers were costing tens of thousands of pounds, the cost of a mechanical keyboard was negligible. But developments in microprocessor technology dramatically reduced the cost of the microcomputer's electronic components.

By the time the Sinclair ZX81 came along, a typewriter-style keyboard could make up a significant proportion of a microcomputer's manufacturing cost. The mechanical moving keyboard, found on models such as the Dragon or the BBC Microcomputer, uses actual switches under the keytops (see illustration on opposite page). When the key is pressed, the internal contacts close to complete a circuit. Switches like this contain numerous components and raise the cost of the keyboard considerably. The BBC computer has 74 keys and some models have more.

One solution to the problem is a new, cheaper type of keyboard. The thinking behind the 'touch-sensitive' keyboard of the ZX81 was that most of the people who bought microcomputers would be mainly interested in playing games and writing small programs.

These activities involve a fairly minimal amount of keyboard work, so it seemed logical that potential micro users would be prepared to settle for a lower quality keyboard. If the advantages of a conventional typewriter-style keyboard could be sacrificed considerable savings could be made.

The ZX81 was designed with a touch-sensitive keyboard, eliminating most of the bits and pieces. This brought the price of the model down, but it didn't provide the ultimate solution. The trouble with a touch-sensitive keyboard is that it doesn't provide much 'tactile feedback' (i.e. you are never quite certain that the key you have just pressed has registered in the computer unless you watch the screen).

Sinclair introduced, for its next product — the Spectrum — the membrane keyboard (see diagram). This kind of keyboard is an improvement, but it still lacks the tactile feedback of the typewriter style.

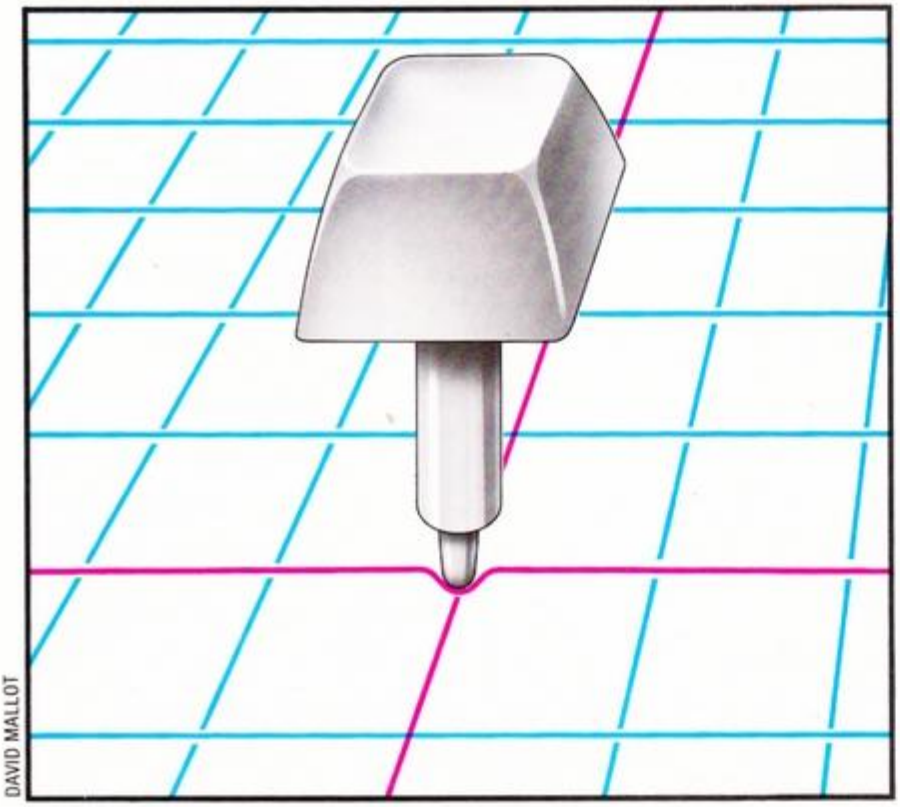
Several relatively low cost computers (including the BBC, the Dragon and the Lynx) have 'professional' typewriter keyboards. The advantages of typewriter keyboards become apparent when the computer is subjected to heavy use for word processing. The familiar typewriter feel enables lots of work to be done quickly.

There is another category of keyboard that lies somewhere between the full moving keyboard and the membrane type of the Spectrum. These are often called 'calculator type keyboards' and are found, for instance, on the NewBrain and the Oric-1. The keys provide a better 'feel' but are small, stiff and are less suitable for touch typing than fully moving typewriter-style keys.

One way of partially overcoming the lack of

The Keyboard Matrix

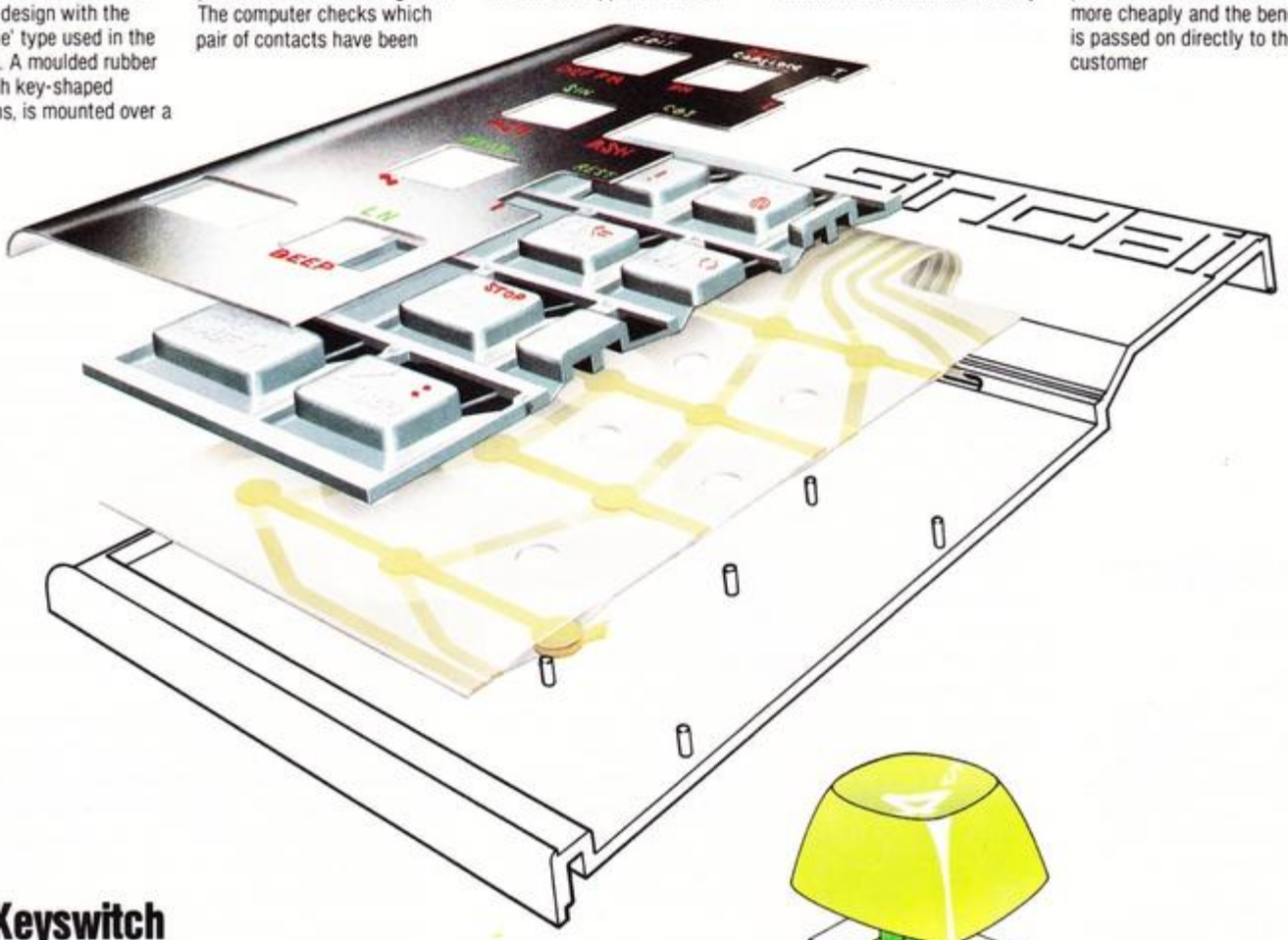
The keys on a computer keyboard are actually switches connected to a grid of wires. The illustration shows how pressing a key connects two wires on the grid. For each key there is one, and only one, pair of wires involved. Each key therefore makes a unique connection on the grid, enabling the computer to figure out which key has been pressed



DAVID MALLOT

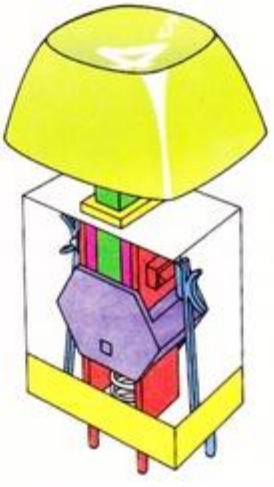
The Sinclair Keyboard

Sinclair advanced the art of keyboard design with the 'membrane' type used in the Spectrum. A moulded rubber sheet, with key-shaped protrusions, is mounted over a pad of contacts forming the keyboard grid or matrix. When a key like this is pressed, a protrusion under the keytop presses the contacts together. The computer checks which pair of contacts have been closed and is able to work out which key this corresponds to. The contacts closed by the key are normally held apart by an air bubble trapped between two plastic sheets. The restoring force to pop the keys back is provided by the elasticity of the the rubber which is stretched when a key is pressed. This original approach to keyboard engineering has certainly cut a few corners but it allows the product to be manufactured more cheaply and the benefit is passed on directly to the customer



The Keyswitch

Typewriter-style keyswitches usually incorporate a pair of contacts. These are normally held apart and do not allow electricity to flow. When the key is pressed a plastic moulding (shown in mauve) moves down and allows the contacts to come together and close a circuit. An internal spring is provided to restore the key to its 'up' position. Closing the contacts allows a current to flow and the computer detects this. The wires connected to the contacts in each switch are arranged in a grid. The computer is able to know which key has been pressed by checking which of the 'vertical' wires on the grid and which of the the 'horizontal' ones is conducting the current. Keys of this type are mechanically complex and manufacturing costs are higher. They offer great reliability and have a more positive 'feel' than rubber membrane keys. This feel or 'touch' comes from the resistance provided by the spring. A well designed key gives a tactile feedback so that the user instinctively knows when the key has been pressed properly. The tops of the keys are also sculpted for more comfortable typing. Keyboards with this type of key are a better choice if the computer is going to be used extensively



DAVID WEEKS

tactile feedback from touch-sensitive and membrane keyboards is to make an audible 'beep' each time a key has been pressed. It reassures the user that the key has actually been pressed and recognised by the computer.

The designers of the Sinclair ZX81 and the Spectrum introduced a novel and very useful way of cutting down on the amount of typing required when entering in BASIC programs. Each key is made to represent more than just a single letter of the alphabet or a number. By using a special 'function' key in conjunction with the ordinary keys, whole BASIC words can be made to appear on the screen without the need to type out the word in full. For example, the BASIC word PRINT can be produced by simply pressing the special function key and the key for the letter P together. A similar idea is used by Sord in the M5 model.

The End of Typing

Not that long ago, the only way to get a computer to do anything was to key an instruction into it by way of the keyboard. This in itself was often a tedious business, made worse if one did not have much skill as a typist. Faced with the fact that these barriers were actually putting people off using (and therefore buying) computers, the manufacturers came up with the brilliantly simple solution of the 'mouse'. The mouse can be shuffled around on any flat surface and, as it moves, the cursor moves around the screen display. One can thus move very rapidly to any part of the screen one wants, press the button, and the operation one wants starts to roll. Mice can also be used in graphics to draw lines or 'paint' colours on the screen

The Electronic Educator

Even the youngest members of your family will be keen to use your computer. Here's the best way to start them off

IAN McKINNELL

One of the most powerful computer aids for primary school children is the floor turtle. This robot is attached to a microcomputer and is operated by a program called LOGO. Children can draw with the floor turtle and it is very useful for teaching mathematical concepts such as shape, distance and the relationship between objects. It is also great fun!

Many parents wonder if a home computer would be of benefit to their children. Most know that it is a good idea for teenage children to learn to use computers at home and at school, but can younger children gain anything from computing?

Yes! The answer is decidedly positive, but there are different ways of introducing a child to the concept of computing, and some are better than others.

The British Government, like many governments in developed countries, has decided that children should now use computers in primary schools. The 'Micros in Primary Education' project is costing £9 million and soon every one of Britain's 29,000 primary schools should have at least one micro. Now the teachers have to find out how to use their small amount of computer power properly.

Computers aren't just good at maths. With a good program — and there is a shortage of good programs for children at present — computers can help young children learn music, ballet, geography, foreign languages and, of course, maths-based subjects such as arithmetic and geometry.

There are two main ways in which a computer can be of help to young children. The child can use the computer to explore his or her world, or the computer can act as a teacher, instructing and drilling the child in a variety of educational subjects.

It isn't a good idea to try teaching your six-year-old how to program a computer in BASIC. Before the age of 12, a child can't really grasp the abstract concepts of such a language. Some children can write programs in BASIC at nine or even earlier, but the work of the French child psychologist Jean

Piaget has shown us that before the age of 12 or 13 most children have trouble grasping abstract ideas.

Reckoning with this problem, researchers have now found a way to let a child control and program a computer without needing to handle such abstract ideas (see the box on LOGO). The usual way teachers introduce children to computers is by a mixture of the two methods.

Turning Turtle

Even very young children can use computers to help them to learn. The picture on the facing page shows a young child playing with a 'turtle', a mechanical robot that is attached to a microcomputer. The turtles are expensive and are intended for school use, but the principle is simple: the turtle has two wheels and a pen. The child tells the turtle to move forward on a piece of paper and tells it whether or not to draw a line — a 'turtle trail' — as it goes. In this way the child draws, instructing the robot how to turn corners and join lines up. Because children are encouraged to work out exactly what moves the turtle has to make to draw a specific shape, they discover for themselves the elements that make up basic geometry. This 'self-help' approach is at the heart of the LOGO method. The belief is that lessons learned 'heuristically' (by trial and error) are better learned than when examples are shown.

These two schools of thought lie at the division in how we use computers with young children. In LOGO, older children, nine or ten perhaps, start using a version of the turtle on the computer screen, drawing intricate shapes and teaching the turtle how to remember various procedures. When a child is 'teaching' the turtle to do things, either on the floor or on the screen, he or she is, in fact, programming the computer. LOGO is a language that allows children to program before they have developed the abstract understanding necessary for most computer languages. Thus 'playing with turtle' allows young children to become used to the idea of controlling the computer and helps them explore their environment.

The other approach uses the apparent 'patience' of the computer to teach children by example.

Children who are having difficulty in understanding a subject or an idea are often helped by 'drill and practice' programs that ask the child questions and then provide a score to show how well he or she has done. Many of these programs are extremely attractive to look at, with good colour graphics and interesting tunes or sound effects. Children are encouraged to learn with these programs and the computer never gets tired or gives up if the child persistently provides the wrong answer. This patience has proved valuable in teaching slow learners, and drill and practice programs that, for example, ask a child to pick a noun out of a group of words, or make up a word out of a set of letters, are very useful educational tools. But to use a computer in this way is to substitute it for the human teacher, and this leads us to an important statement: no computer can replace the human teacher. Human contact is the most important element in teaching and, although a computer is the most powerful educational aid there is, it is no substitute for caring instruction.

If you are thinking of buying a computer for your children to use, it is worth finding out which type of computer is in use at your child's school. Buying a similar model will allow your children to use the same programs at home and will forge a link between computer activity at school and in the home.

Computers are fun, and it is quite all right to allow children to play games with them. A lot of parents are worried about the possible addictive effects of games such as Space Invaders or Pac-Man, but although these games are particularly enjoyable, there is no evidence whatsoever to suggest that their appeal goes beyond fascination.

Under seven, children need help and supervision in turning on a computer and television set and loading a program. If the program is good

LOGO Logic

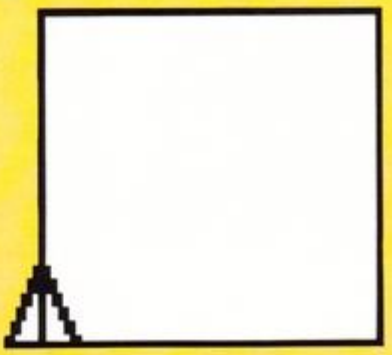
Here we show how shapes are built up on the screen using the LOGO language.

LOGO is a computer language developed specifically to allow young children — as young as four or five — to program a computer. It was developed at the Massachusetts Institute of Technology in the late 1960s by a team led by Seymour Papert, a mathematician who had worked with the world-famous educationalist Jean Piaget at his Geneva Centre.

For the youngest children, LOGO takes the form of a 'turtle', that is either a mechanical robot on the floor or a triangle of light on a computer screen. The command FORWARD 10 causes the turtle to move forward 10 units, drawing a line behind it. The command RIGHT 90 causes the turtle to make a right angle. Chains of commands can be built up that cause the turtle to draw squares, triangles, circles and unorthodox shapes as well. The turtle can be also taught to 'remember' the commands. Without realising it, children teaching a turtle are, in fact, programming a computer

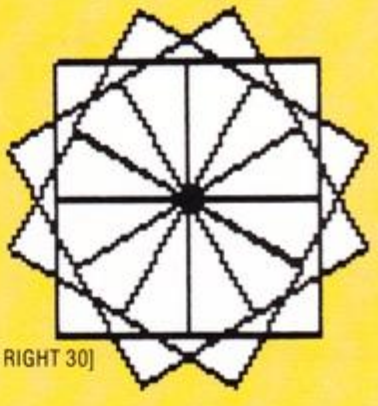
One way to draw a square box:

```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
```



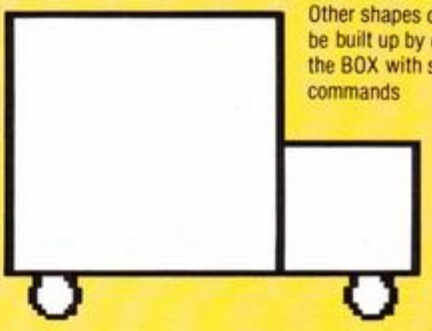
Constructing a 'BOX' command:

```
TO BOX
REPEAT 4 [FORWARD 50 RIGHT 90]
END
```



The STAR command:

```
TO STAR
REPEAT 12 [BOX RIGHT 30]
END
```



Other shapes can easily be built up by combining the BOX with similar commands

Eye Strain

Some adult workers staring at computer screens do suffer from eye-strain and fatigue, but children don't have long enough concentration spans for this to become a problem. The problem for adults seems to be 'accommodative lock', a phenomenon in which the eye gets fixed at one focal length and takes a time to re-adjust. If you think that your child might be so keen on computing that he or she will stare solidly at the screen for long periods, problems can be avoided by ensuring that a short break is taken every 15 minutes.

One technical problem about old domestic televisions needs to be mentioned. It has been found that a few colour televisions made before 1970 are capable of emitting a low radiation dose when used regularly for close-up work that could be dangerous. If you are considering using an old colour television set for computing purposes it is worth ensuring that it was made after 1970

enough, they may then be left to use it, although this depends on their reading skill and ability to provide answers to the program's questions. Hardware requirements are pretty simple. A microcomputer has to be robust: children *do* hit the keyboard with clenched fists, they pull at the electrical connections and they constantly touch and prod the screen. If a system is flimsy, poorly connected or hard to use, it will fail to interest a young child. Some experts consider that a computer keyboard for a very young child must have large, clearly defined keys. But as children's motor movements become fully developed (usually by the age of seven), they are capable of dealing with 'fiddly' keyboards that would seem difficult even to adults. The touch-sensitive keyboards found on the cheapest Sinclair microcomputers are not really suitable for children under nine or ten, although the larger versions of the printed keyboard, such as the one on the Philips Videopac 7000, are suitable for four-year-olds. It is a question of size and ease of use.

Young Programmers

Software choice is more difficult for young children. If you intend using a cassette-based system you will have to supervise all loading and storage. If your system is disk-based you will find that young children are able to handle floppy disks very well. One of the best forms of program storage for the very young, those under seven, is the ROM cartridge, a plastic case that contains a chip with an electrically-embedded program. The disadvantage of such a system is that it doesn't allow the user to store any work, but cartridges are virtually indestructible and allow young children to use computers without any supervision.

If you make the decision to buy a computer specifically for your children, do try to provide it with a permanent home. Moving a computer from room to room, with the connection and disconnection of leads that involves, won't do a computer any harm (unless you drop it), but it is likely to l skipped by the child in favour of something le tedious — like watching television.

The Ideal Work Station

In an ideal world, the child's computer shou be set up in his or her bedroom complete with i own television screen. If you are serious abo your children developing a positive attitude computers, consider setting up a work-station one of their bedrooms and supplying a secon hand television exclusively for their use. (Put t computer centre in the oldest child's bedroom. H or she may want to use it after the others have go to bed.) Old black-and-white televisions can l purchased very cheaply and, provided there is facility for channel tuning, they are perfectly ca able of displaying computer information. There a great deal of argument about the value of colo in computing for young children; some experts s it is vital, others consider it a bonus, but u necessary. It seems obvious that if the choice between a permanent connection to a black-an white television set in a bedroom, or a tempora connection to the family's colour television, t permanent set-up is infinitely more desirabl

If you are able to set up a permanent, or sem permanent, computer work/play centre in one the children's bedrooms, it is a good idea arrange things so that the computer can b removed without disturbing the set-up. Whe organising the computer table or bench, tape dow all the leads and connections so that the childre won't accidentally pull out a lead. (Ensure that a mains connectors are safely protected and tape down in such a way that they cannot becom unsafe. And don't supply an aerial connection the television, otherwise the late film may 'accide tally' appear on the screen after bed time.) It important that the microcomputer is stable a doesn't bounce around. Sinclair can supply a tr that holds its very light Spectrum computer stea and, if your children are boisterous, you mig consider building a clamp or other means of hol ing the microcomputer down. Of course, the co puter you buy for your children will also be of u to the whole family so, if the family is sharing o computer, it is worthwhile buying duplicate lea and (if necessary) a second mains supply pack the computer. These are relatively inexpens and they will allow you to say a firm good night your kids, unplug the computer and cassette corder (or disk drive) from their bedroom cen (leaving all leads taped in place) and scu downstairs to plug them into your own televis using your duplicate leads. If a computer has to set up and taken down in the living room whene your children wish to use it, the fuss of lead c nections and the possible disputes with memb of the family who would prefer to watch an ea evening soap opera may, unless carefully c trolled, kill the idea of computing as a leis activity for children even before it begins.

The Big Trak

It looks like a toy tank, but in fact it's a powerful learning tool. The Big Trak is a computer-based programmable toy that allows a child to plan out precisely what moves he or she wants the tank to do. The tank can remember up to 16 steps and may be programmed to wander from room to room around the house before returning to base. The child has fun, but the computer is helping him or her to explore the physical world and work out the individual steps necessary in a simple computer program. Despite its aggressive looks, girls love Big Trak as much as boys do!

Looping The Loop

Breaking out of loops, going round them a required number of times, and line numbering are covered in the second part of our programming course

We ended the last part of the 'Basic Programming' course with the program listed below. It worked fine but because of the GOTO in line 70 the program kept looping back to the beginning and never stopped. The only way to get out of the loop was to use the BREAK key or the RESET key.

Now we are going to look at one of the ways we can get out of a loop like this by incorporating a test in the program. The usual way it's done is to test for a number we would never actually want to use in the program. The program allowed us to type in a number that the computer then printed on the screen with a 1 added to it. We might decide that we would never want to enter a number bigger than 999. In that case we could test to see if the number that has been input is greater than 999. Type in the program and then add:

35 IF A > 999 THEN GOTO 80<CR>

Now run the program again and it will function as before — unless, that is, you enter a number greater than 999. Try typing 1000<CR> and see what happens.

Why did the program stop this time? The IF in line 35 is what made it happen. When BASIC finds an IF statement it knows that a logical test is coming. The > sign means 'greater than'. Line 35 therefore means IF (variable) A (is greater than) 999 THEN GOTO (line) 80. If you just typed in 1000, the value of A becomes 1000 which is greater than 999 so the program THEN GOes TO line 80 which is the end of the program. If A is not greater than 999, the THEN part of the line is ignored and the program continues to the next line.

When running this program, then, you can input numbers as often as you like, just as long as they are not greater than 999. As soon as a number bigger than 999 is input, the IF-THEN statement detects the fact and terminates the program by GOing TO the END. When a BASIC program has reached the end or been terminated, you will be given a 'ready' prompt on the screen. Depending on your computer, this prompt may take several forms. On the BBC Microcomputer the ready prompt is a sign like this: >. On the Dragon it's OK. On the Sord it's READY. Whatever form it takes, the ready prompt is BASIC's way of telling you that no program is running and that it is awaiting further orders.

There is a lot of variation in the way different versions of BASIC use THEN. Details are given in the 'Basic Flavours' box on page 39.

Other comparisons used in BASIC are < (less than), = (equals), >= (greater than or equal to), <= (less than or equal to) and <> (not equal to). We'll see these comparisons used often as the course progresses.

Before continuing any further, it's worth trying out a few exercises to get the feel of using these comparisons.

Exercises

- Change one of the lines so that the program will be aborted if A = 1000.
- Change one of the lines so that the program will be aborted if the number input is less than zero.
- Change the GOTO line so that it makes the program loop back to the beginning if A is equal to or less than 500. Hint: you will not need a separate IF-THEN line and a GOTO line.

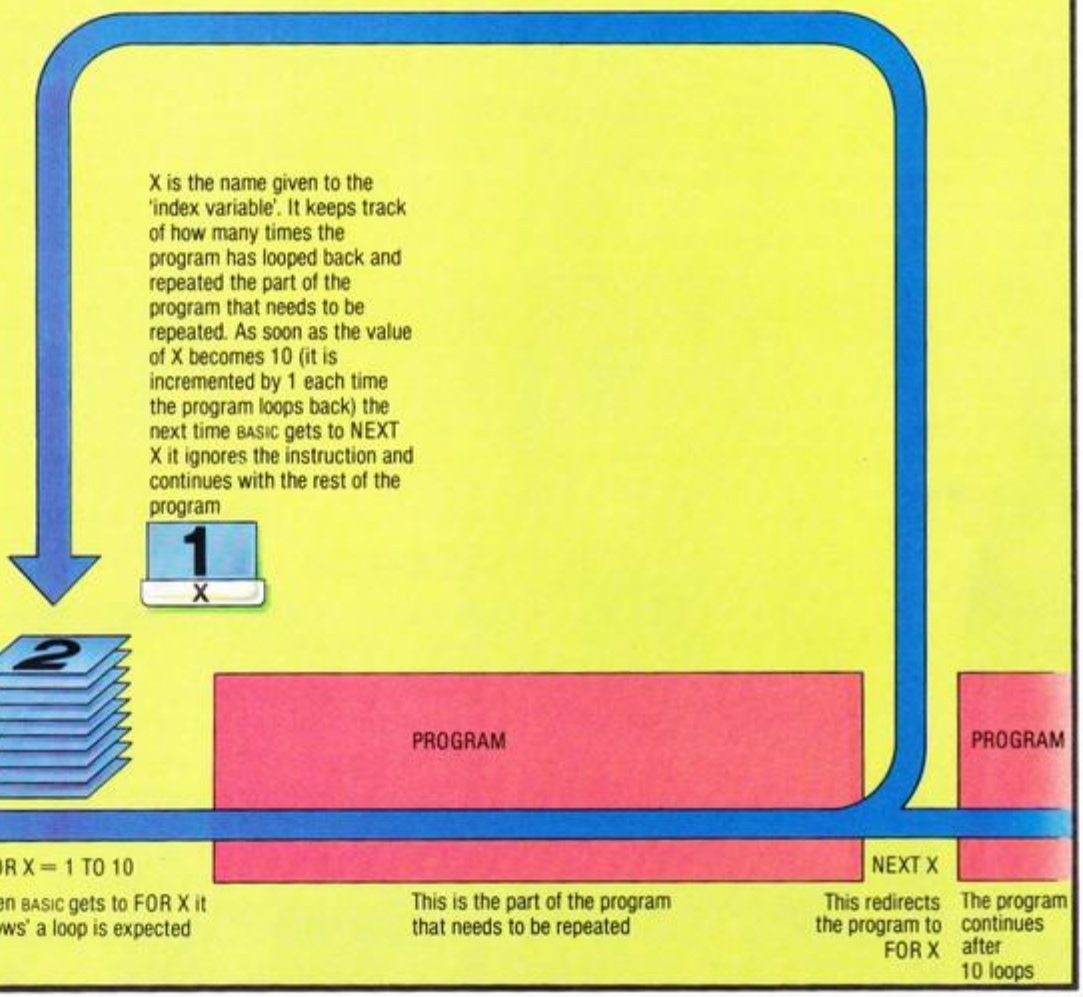
Discovering FOR-NEXT

When writing programs there will be many occasions when you would like some items in the program to be repeated a precise number of times. The GOTO in line 70 enabled the program to loop as many times as we wanted. We later added an IF-THEN statement in line 35 which enabled us to escape by entering an 'out of range' number.

```
10 REM COMPUTERS NEVER MAKE MISTAKES
20 PRINT "TYPE IN A NUMBER"
30 INPUT A
40 LET A = A + 1
50 PRINT "I THINK THE NUMBER YOU TYPED WAS ";
60 PRINT A
70 GOTO 20
80 END
```

The FOR-NEXT Loop In Basic

The FOR-NEXT statements in BASIC creates a loop so that part of the program can be repeated an exact number of times. When BASIC encounters the word FOR, it knows that the next part of the program will be repeated. It creates a variable (X in this case) and sets its values to 1. The next part of the program (the part to be repeated) is 'executed'. When BASIC encounters the NEXT X statement, instead of continuing with the rest of the program, it loops back to the FOR X line, increments the variable X (pictured here as the next card being popped into a card holder labelled X) and then carries on with the 'loop' part of the program again. This process is repeated 10 times


X is the name given to the 'index variable'. It keeps track of how many times the program has looped back and repeated the part of the program that needs to be repeated. As soon as the value of X becomes 10 (it is incremented by 1 each time the program loops back) the next time BASIC gets to NEXT X it ignores the instruction and continues with the rest of the program
1
X
2
PROGRAM
PROGRAM
FOR X = 1 TO 10
NEXT X
This is the earlier part of the program
When BASIC gets to FOR X it 'knows' a loop is expected
This is the part of the program that needs to be repeated
This redirects the program to FOR X
The program continues after 10 loops

However, there are occasions, as we learnt in the first part of the course, when using GOTO to make a loop is not the best way of doing things.

Let's return to our old program, modified now to tell the truth this time, to multiply the number input by 10, and do it exactly eight times.

```
10 REM MULTIPLY BY 10
20 FOR X = 1 TO 8
30 PRINT "TYPE IN A NUMBER"
40 INPUT A
50 LET A = A * 10
60 PRINT "YOUR NUMBER MULTIPLIED BY 10
   IS ";
70 PRINT A
80 NEXT X
90 END
```

Type this program in, LIST it to check for mistakes and then RUN it. You will be asked for a number only eight times. After that the program simply stops. The reason this happens is to be found in line 20.

```
20 FOR X = 1 TO 8
```

This is part of a FOR-NEXT loop. It is one of the most useful structures BASIC has to offer. It deserves careful study.

The way we have used it here, we have created a variable called X. (Variables are explained in the first part of the course on page 21.) We could have called it anything (except A — which we are using for something else). FOR must always be used with a corresponding NEXT, but the NEXT will appear later in the program — after the portion to be repeated. The FOR part of a FOR-NEXT loop always has the following form:

FOR variable = starting value TO final value

In our example FOR X = 1 TO 8 we have called the variable X and given it an initial value of 1. The next part of the program is then executed by the computer; the number we typed in is multiplied by 10 and then printed on the screen. After that we get to NEXT X and the program loops back to where variable X is — in line 20. As soon as it has done that it increments X by 1, so X acquires a value of 2. The part of the program within the FOR-NEXT loop is then executed again. On coming to NEXT again in line 80, the program loops back and increments X to 3.

The program continues to repeat like this until X has been incremented to 8. After that, the loop is terminated; NEXT X does not go back to FOR X and the program continues to the next line.

More Uses For FOR-NEXT Loops

FOR-NEXT loops are often used to create delays in the program. There are times when you don't want everything done at maximum speed and so you introduce a delay. You probably found that the answers in the MULTIPLY BY 10 program flashed up so quickly they seemed instantaneous. Let's make the computer look as if it's having to think before it answers by using FOR-NEXT to insert a delay. Add the lines shown in blue type to your program.

```
10 REM MULTIPLY BY 10
20 FOR X = 1 TO 8
30  PRINT "TYPE IN A NUMBER"
40 INPUT A
50 LET A = A * 10
52 FOR D = 1 TO 1000
54 NEXT D
60 PRINT "YOUR NUMBER MULTIPLIED BY 10
   IS ";
70 PRINT A
80 NEXT X
90 END
```

We have added another two lines, 52 and 54, inside our original FOR-NEXT loop. Let's look at them.

```
52 FOR D = 1 TO 1000
54 NEXT D
```

D is set to 1 and the program goes to the next line. This is the corresponding NEXT statement. Nothing actually happens inside the loop, the program simply loops back to line 52 and increments D to 2. This happens 1000 times before the program goes to the next part — which is printing the answer. Computers are fast, but everything takes a finite time, so looping back 1000 times takes a noticeable amount of time. Computers vary in the time they take to loop. On the Epson HX-20 this FOR-NEXT loop takes 2.9 seconds, while on the Spectrum it takes 4.5 seconds. Experiment by changing the number you use as the upper limit in line 52.

To make the computer behave more like a human being, add these three lines:

```
56 PRINT "NOW LET ME SEE..."
57 FOR E = 1 TO 1000
58 NEXT E
```

LIST the program and RUN it. We now have two delays that do absolutely nothing except waste time.

Add these two lines:

```
51 REM THIS LOOP WASTES TIME
55 REM THIS WASTES MORE TIME
```

Now LIST the program and have a good look at it. Notice how all the extra lines we have added have fitted into exactly the right places. Which brings us to the last point in this instalment of the course — line numbers.

We started our original program with line 10 and went up in jumps of 10 for each new line, ending with line 90. We could have chosen any numbers, for example 1, 2, 3 . . . 9. But if we had done that, how would we have fitted in the extra lines? Programmers always have afterthoughts and improvements to make, so allow for these by leaving big gaps between line numbers in the 'Mark I' versions of their programs. You could even start with line number 100 and go up in jumps of 50 or 100 if you wanted.

Some versions of BASIC include a useful command called AUTO. BBC BASIC has it, so does the Epson HX-20. The Dragon, Sinclair computers and the VIC 20 do not. If your BASIC has AUTO you can save a lot of time by having the line numbers generated for you automatically. Find out if your BASIC has AUTO by typing:

```
AUTO 100, 10<CR>
```

If your BASIC does have AUTO you will see on the screen:

```
100
```

The screen shows the number 100 followed by a space and then the *cursor*. The cursor is a mark (sometimes a line, or a square) that shows on the screen where the next character will appear. You can start entering the first line of the program from the cursor position. When you hit <CR> the next line will appear automatically, starting with the line number 110. AUTO, if you have it, can either be used by itself, or with one or two 'arguments'. *Argument* is a mathematical term. In the expression 2 + 3 = 5, the arguments are 2 and 3. With the AUTO command, it can be used just by itself (i.e. AUTO<CR>) or with one 'argument' (e.g. AUTO 100<CR>) or with two arguments (e.g. AUTO 300,50). AUTO by itself usually causes line numbers to start with 10 and to go up in increments (jumps) of 10. If only one argument is used (e.g. AUTO 100<CR>) the first number will be 100 (in this case) and then the numbers will go up in the 'default value' — which again is usually 10. If you specify two arguments, the first number specifies the starting line number and the second number specifies the increment. AUTO 250,50<CR> will give a starting number of 250, the next number will be 300 and so on in increments of 50. Even on the simplest micro, you're unlikely ever to run out of lines.

In the next instalment of this course we will look at various ways of improving the visual presentation of the program on the screen and different ways of printing out data.

Basic Flavours

IF — Most microcomputers can use this instruction in the form of either IF A > 999 THEN 80 or IF A > 999 GOTO 80. (The Spectrum uses IF A> 999 THEN GOTO 80.)

AUTO — This command is not available on the Commodore VIC 20, DRAGON 32 or Sinclair Spectrum.

Behind The Bars

How those mysterious-looking stripes on book covers and supermarket goods spell out a message to a computer and help to run the store more efficiently

Decoding The Stripes

The illustration above shows a bar code. This represents the number 72. It appears as nothing more than a series of black lines of different thicknesses. In this case each set contains five bars, two of which are broad. The position of the two broad lines in each set gives the number. Extra bars mark the beginning and end of each individual unit of information. Here the number 72 is coded in two units, the digit 7 and the digit 2.

There are many different ways of encoding information in a bar code. Since a bar is either thick or thin, the bar can be numerically represented as a 1 or a 0. This leads directly into the binary mathematics of computers. A somewhat different bar code in commercial use is the Universal Product Code. Here the bars can be of variable thicknesses. Fewer bars are needed, and the information is read from the breadth of the line.

It is because bar codes can be read by machine that they have become so widespread in supermarkets and libraries. The can of Coke goes past a light detector, and the librarian wields a light-sensitive 'pen'. In both cases the bar code is illuminated and the amount of light reflected is registered. The black bars reflect hardly any light in comparison with the white background. The reflected light is converted into an electrical signal and amplified. Either light is registered (a binary 1) or there is no light and no signal (a binary 0). The information is now in a form that the computer can accept.

The bars give strings of 0s. The wider the bars are, the more 0s they contain. Similarly, the white background gives strings of 1s. In this way, the wand feeds the computer patterns of binary digits from which it can determine the composition of a bar code

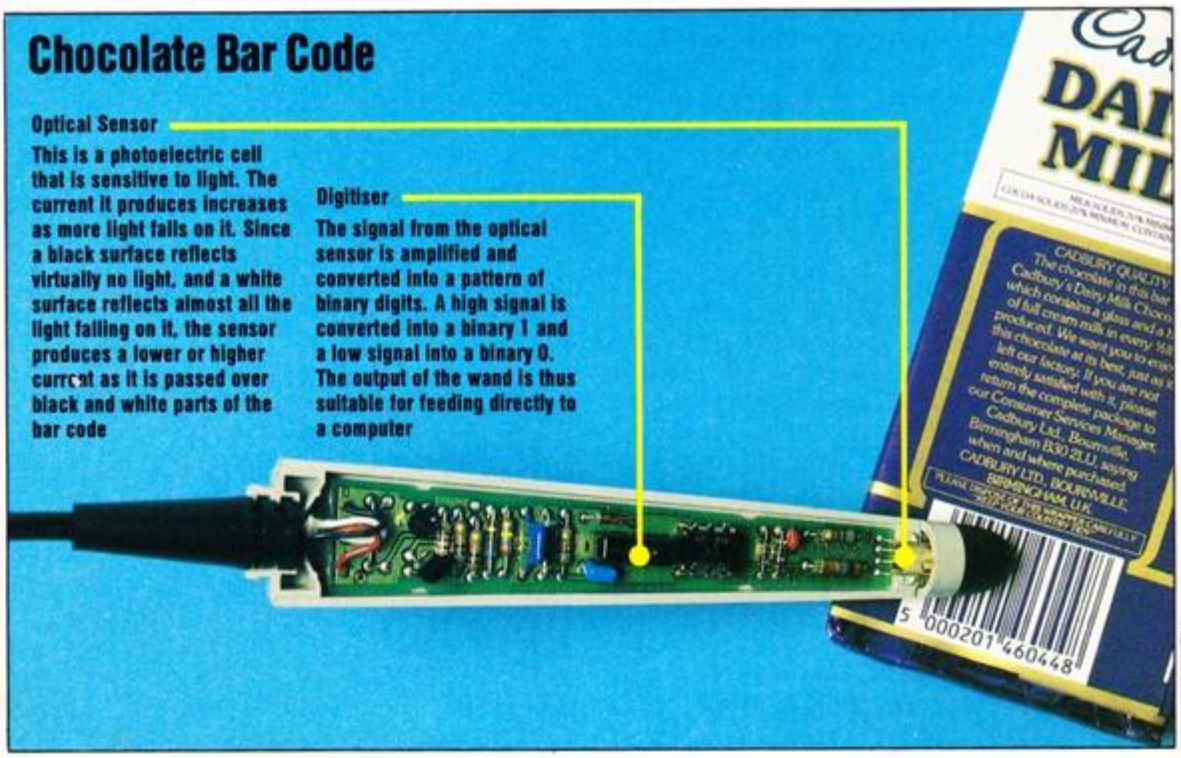
Noticed anything different about the cans of Coke you've been buying lately? Or about a lot of paperback books? Or maybe you've noticed a curious kind of trolley being wheeled around your local branch of Sainsbury's, with someone running a kind of pencil (making blips and beeps) along the shelves. Take a close look at the shelves, or the Coke can or a recently published paperback, and you'll see a set of stripes. That is a bar code, an ingenious device that can be read in a fraction of a second by a light-sensitive 'wand', and that can feed information about the goods involved directly into a store's computer. That, in turn, means that information about cash flow, stock levels and so on is instantly available to the people who keep the shelves full, who run the warehouse and who buy in the goods in the first place — all making for a much more efficient service for the customer.

Let's see how this works for a paperback. Every book published in all the major countries of the world has an International Standard Book Number (ISBN). This consists of one or more digits to indicate the language or the geographical area in which the book is published (it's 0 for all English language books), from two to seven digits to identify the publisher, and from one to six digits to identify the individual book title and edition. This gives a total of nine digits — then there's a check digit (which the computer uses to make sure that all these digits have been given to it in the correct order).

For bar-coding, books are numbered according to the European Article Numbering (EAN) system, which uses a total of thirteen digits (most grocery items usually employ a short eight-digit number). The first three digits are the EAN 'flag' — 978 for books. Then comes the ISBN, and finally an alternative EAN check digit. In the USA and Britain, the ISBN (complete with its own check digit) is also printed above the bar code in figures that can be read both by the human eye and by an Optical Character Reader.

Optical character readers are another interesting development with far reaching consequences. Machines now exist which can literally read the printed word by scanning the line optically. The output signal of the reader is coupled to the computer, which can then process the information in various ways. Words read by the scanner could, for example, be displayed on the computer screen, avoiding the need for laborious typing and all thanks to the computer.

Chocolate Bar Code


Optical Sensor
This is a photoelectric cell that is sensitive to light. The current it produces increases as more light falls on it. Since a black surface reflects virtually no light, and a white surface reflects almost all the light falling on it, the sensor produces a lower or higher current as it is passed over black and white parts of the bar code
Digitiser
The signal from the optical sensor is amplified and converted into a pattern of binary digits. A high signal is converted into a binary 1 and a low signal into a binary 0. The output of the wand is thus suitable for feeding directly to a computer
CADBURY QUALITY
CADBURY LTD., BOURNVILLE, BIRMINGHAM, U.K.
5 000201 460448

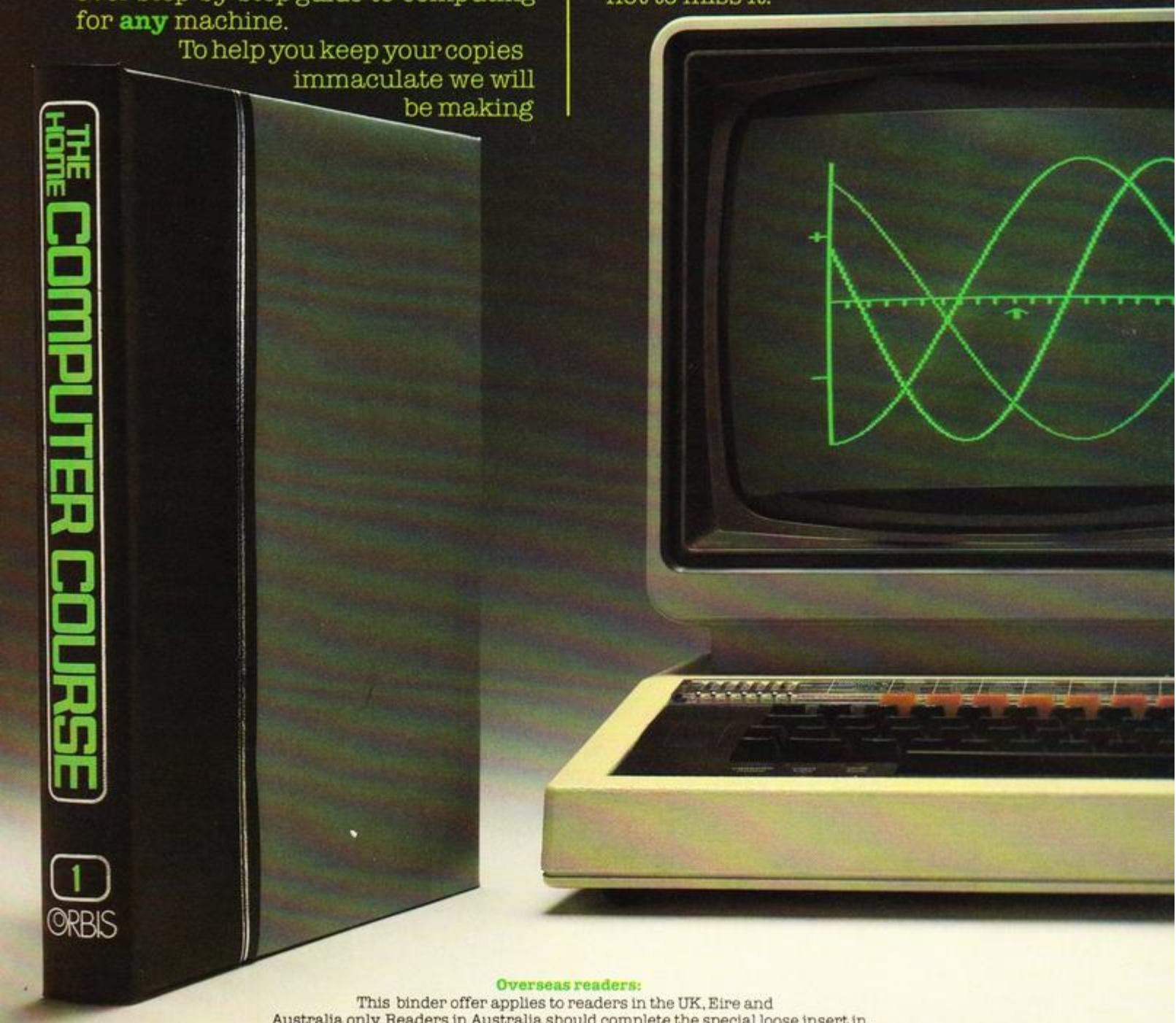
IAN MCKINNELL

THE HOME COMPUTER COURSE BINDER.

A BASIC NECESSITY.

The Home Computer Course grows week-by-week into the first ever step-by-step guide to computing for **any** machine.

To help you keep your copies immaculate we will be making a very special binder offer with an order form in a future issue. Be sure not to miss it.

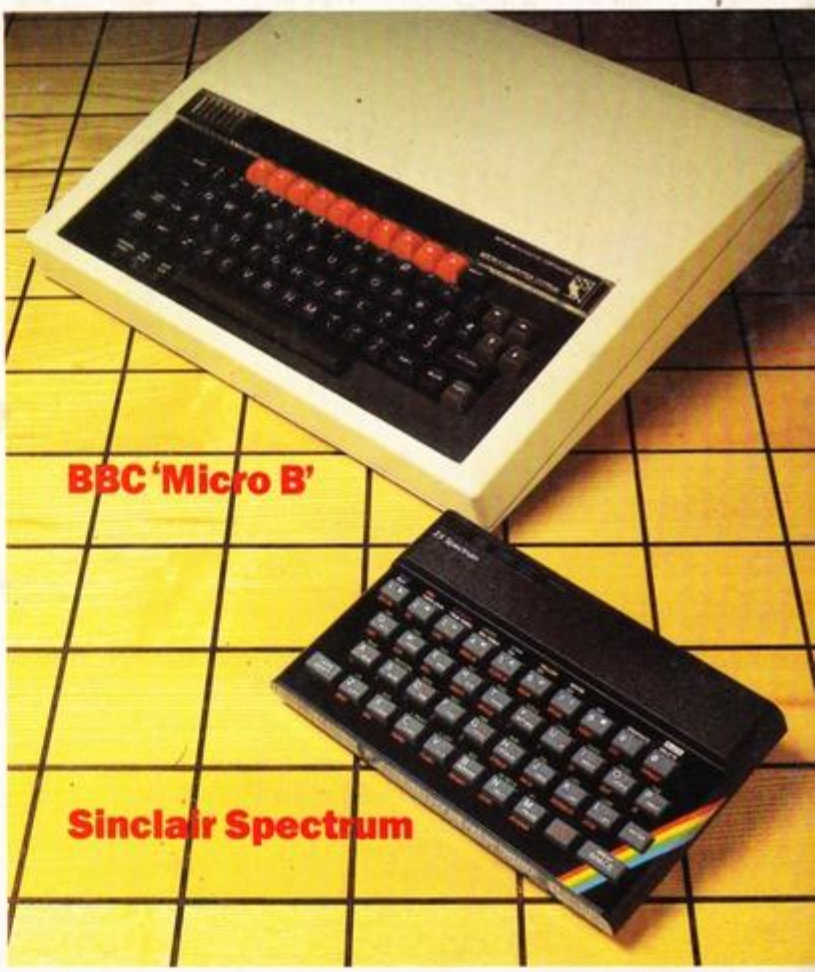


Overseas readers:

This binder offer applies to readers in the UK, Eire and Australia only. Readers in Australia should complete the special loose insert in Issue 1 and see additional binder information on the inside front cover. Readers in New Zealand and South Africa and some other countries can obtain their binders **now**. For details please see inside the front cover.

Binders may be subject to import duty and/or local tax.

WIN A BBC 'MICRO B' HOME COMPUTER



10 First prizes of BBC 'Micro B' computers

PLUS

50 Sinclair Spectrums for runners-up

Just look at what the 'Micro B' has to offer.

- Recommended by the Government for use in primary and secondary schools.
- Uses the BBC version of the most popular computer language – BASIC.
- Loaded from any standard audio-cassette recorder.
- Has 73 key full typewriter-style keyboard with 10 easy-to-understand function keys.
- 32 K Random Access Memory (RAM) gives high definition graphics.
- 3 part music synthesiser capability.
- Standard in-built connectors for domestic T.V./monitor/audio-cassette/computer accessories.
- Easily adaptable to other computer languages (e.g. Pascal, FORTH and LISP).
- Flexible additions include word processor, disk drive and Micronet/ Prestel adaptors.

HOW TO ENTER:

The official entry form will appear in issue 4. Issue 3 will contain a free glossary of computer terms which sh help you with your answers to the definitions below. In the meantime, we have set out below what you will h to do.

Remember, you will need issue 4 for the official entry fo

ALL YOU HAVE TO DO:

1. Give the answers to the following definitions

A. A group of four bits, or half a byte.

B. A collection of bits (usually eight) that make up the smallest amount of usable information in a program.

C. An impact printer mechanism with the letters and numbers on spokes attached to a central stem.

D. A set of instructions that are frequently required during the running of a program and that can be called up at any point in the program.

E. A number system on a base of 16 that is written in both letters and numbers.

F. A correction made to a computer program, usually as a group of instructions added to correct a mistake.

2. Complete the following statement in not more than 10 words:
"In only 24 weeks, The Home Computer Course will make me _____"

RULES

1. By entering the competition, competitors will be deemed to have accepted and agreed to abide by the rules.
2. The competiton is open to all UK and Eire readers other than employees or their families of Orbis Publishing Ltd. and their advertising and servicing agents.
3. All valid entries will be examined.
4. The first prize will be awarded to the competitor who selects, in the judges' opinion, the most suitable answers to the questions listed, in the event of more than one competitor qualifying for the first prize this will be awarded to the qualifying competitor who, in the judges' opinion, submits the most apt answer to the special tie-breaker question in issue 4. A similar basis will be adopted for determining the winners of the runner-up prizes. No house-hold may win more than one prize.
5. All entries must be ink or ball-point pen.
6. The closing date for entries will be given in issue 4. Winners will be notified by post and a full list of winners will be available for inspection at 20-22 Bedfordbury, London WC2, one month after the closing date.
7. No responsibility will be taken for entries lost, delayed or damaged in transit. Proof of posting cannot be accepted as proof of delivery.
8. Illegible entries and entries not mad in accordance with the rules and directions will be disqualified.
9. All entries submitted will become the copyright of Orbis Publishing Ltd. an no entries can be returned.
10. The judges' decision is final and legally binding, and the decision of Orbis Publishing Ltd. on all other matters concerning the competition will also be final and legally binding. No correspondence will be entered into.

This competition is open to readers in the UK and EIRE only.